目　录

第1章 绪　论

1.1 序言

人类传递信息的主要媒介是语音和图像。据统计，在人类接受的信息中，听觉信息占20%，视觉信息占60%，其他如味觉、触觉、嗅觉总的加起来不过占20%。所以，作为传递信息的重要媒体和手段——图像信息是十分重要的，俗话说"百闻不如一见"、"一目了然"，都反映了图像在传递信息中的独到之处。从技术发展来看，图像通信却大大落后于语音通信，语音通信发展速度远比图像通信来得快，自从Bell发明电话以来，话音通信发展十分迅速，如：交换技术从人工交换到自动交换，交换机从步进制、纵横制到目前的程控交换机，发展相当快，而目前传统的电话通信又受到了IP电话的挑战。尤其是最近几年我国的通信事业发展速度之快是惊人的，已基本与世界水平接轨了。近年来，移动通信的发展亦十分迅速，蜂窝网、集群电话、无绳电话、无线寻呼，GSM、W-CDMA等都获得了巨大成功，IMT2000也正在迅速实施之中，个人通信的目标已离我们越来越近。实际上，目前的蜂窝系统也是一种个人通信系统，尤其是具有漫游功能的手机就是一种个人通信方式，而最终发展的是个人号码式个人通信。电话机也可以做得很小，如卡片式，甚至可做成手表式或戒指式的电话机。近年来IP电话对传统的电话系统提出了强烈挑战，大有取而代之之势，其优点是明显的。

相比之下，图像通信就大大落后于语音通信了。虽然电视广播已普及到千家万户，但类似于电话通信网那样的图像通信网尚没有建立。尽管近年来伴随着多媒体通信的发展，可视电话、会议电视已有了较大的发展，相应的国际标准也纷纷制定出来了，也有了一些专业产品，但总体来看尚不够普遍，而且大都是在专业网下实现的图像通信，至于图像处理技术是在第三代计算机问世后才得到了迅速发展。目前，图像处理技术发展迅速，其应用领域也愈来愈广，有些技术已相当成熟并产生了惊人的效益。当前图像处理面临的主要任务是研究新的处理方法，构造新的处理系统，开拓更广泛的应用领域。

数字图像处理技术起源于20世纪20年代，当时通过海底电缆从英国伦敦到美国纽约传输了一幅照片，它采用了数字压缩技术。就1920年的技术水平来看，如果不压缩，传一幅图像要一星期时间，压缩后只需要了3小时。1964年美国的喷气推进实验室处理了太空船"徘徊者七号"发回的月球照片，这标志着第三代计算机问世后数字图像处理概念开始得到应用。其后，数字图像处理技术发展迅速，目前已成为工程学、计算机科学、信息科学、统计学、物理学、化学、生物学、医学甚至社会科学等领域各学科之间学习和研究的对象。如今图像处理技术已给人类带来了巨大的经济和社会效益。不久的将来它不仅在理论上会有更深入的发展，在应用上亦是科学研究、社会生产乃至人类生活中不可缺少的强有力的工具。

图像处理科学对人类具有重要意义，它表现在如下三个方面：

（1）图像是人们从客观世界获取信息的重要来源

人类是通过感觉器官从客观世界获取信息的，即通过耳、目、口、鼻、手通过听、看、味、嗅和

触摸的方式获取信息。在这些信息中,视觉信息占60%～70%。视觉信息的特点是信息量大,传播速度快,作用距离远,有心理和生理作用,加上大脑的思维和联想,具有很强的判断能力。其次是人的视觉十分完善,人眼灵敏度高,鉴别能力强,不仅可以辨别景物,还能辨别人的情绪,由此可见,图像信息对人类来说是十分重要的。

(2) 图像信息处理是人类视觉延续的重要手段

众所周知,人的眼睛只能看到可见光部分,但就目前科技水平看,能够成像的并不仅仅是可见光。一般来说可见光的波长为0.38～0.8μm,而迄今为止人类发现可成像的射线已有多种,如:

γ射线: 0.003～0.03nm;

X射线: 0.03～3nm;

紫外线: 3～300nm;

红外线: 0.8～300μm;

微波: 0.3～100cm。

这些射线均可以成像。利用图像处理技术把这些不可见射线所成图像加以处理并转换成可见图像,实际上大大延伸了人类视觉器官的功能,扩大了人类认识客观世界的能力。

(3) 图像处理技术对国计民生有重要意义

图像处理技术发展到今天,许多技术已日趋成熟。在各个领域的应用取得了巨大的成功和显著的经济效益。如在工程领域、工业生产、军事、医学以及科学研究中的应用已十分普遍。通过分析资源卫星得到的照片可以获得地下矿藏资源的分布及埋藏量;利用红外线、微波遥感技术可侦查到隐蔽的军事设施; X射线CT已广泛应用于临床诊断,由于它可得到人体内部器官的断层图像,因此,可准确地确定病灶位置,为诊断和治疗疾病带来了极大的方便。至于在工业生产中的设计自动化及产品质量检验中更是大有可为。在安全保障及监控方面图像处理技术更是不可缺少的基本技术,类似的应用例子随处可见。至于在通信及多媒体技术中图像处理更是重要的关键技术。因此,图像处理技术在国计民生中的重要意义是显而易见的。正因为如此,图像处理理论和技术受到了各界的广泛重视,科学工作者经过不懈的努力,已取得了令人瞩目的成就,并正在向更加深入及更高的层次发展。

1.2 图像处理技术的分类

图像处理技术基本可分为两大类: 模拟图像处理和数字图像处理。

1.2.1 模拟图像处理

模拟图像处理(Analog Image Processing)包括: 光学处理(利用透镜)和电子处理,如:照相、遥感图像处理、电视信号处理等。模拟图像处理的特点是速度快,一般为实时处理,理论上讲可达到光的速度,并可同时并行处理。电视图像是模拟信号处理的典型例子,它处理的是活动图像, 25帧/秒。模拟图像处理的缺点是精度较差,灵活性差,很难有判断能力和非线性处理能力。

1.2.2 数字图像处理

数字图像处理(Digital Image Processing)一般都用计算机处理或实时的硬件处理,因此也

称之谓计算机图像处理(Computer Image Processing)。其优点是处理精度高,处理内容丰富,可进行复杂的非线性处理,有灵活的变通能力,一般来说只要改变软件就可以改变处理内容。其缺点是处理速度还是一个问题,特别是进行复杂的处理更是如此。一般情况下处理静止画面居多,如果实时处理一般精度的数字图像需要具有100Mips的处理能力;其次是分辨率及精度尚有一定限制,如一般精度图像是512×512×8bit,分辨率高的可达2048×2048×12bit,如果精度及分辨率再高,所需处理时间将显著地增加。

广义上讲,一般的数字图像很难为人所理解,因此,数字图像处理也离不开模拟技术,为实现人-机对话和自然的人机接口,特别需要人去参与观察和判断的情况下,模拟图像处理技术是必不可少的。

1.3 数字图像处理的特点

数字图像处理的特点表现在如下几个方面:

(1) 图像信息量大

在数字图像处理中,一幅图像可看成是由图像矩阵中的像素(pixel)组成的,每个像素的灰度级至少要用6bit(单色图像)来表示,一般采用8bit(彩色图像),高精度的可用12bit或16bit。一般分辨率的图像像素数为256×256、512×512,高分辨率图像像素数可达1024×1024或2048×2048。

例如:256×256×8≈64kB

512×512×8≈256kB

1024×1024×8≈1MB

2048×2048×8≈4MB

X射线照片一般用64~256kb的数据量,一幅遥感图像3240×2340×4≈30Mb,因此,大数据量给存储、传输和处理都带来巨大的困难。

(2) 图像处理技术综合性强

在数字图像处理中涉及的基础知识和专业技术相当广泛。一般来说涉及通信技术、计算机技术、电子技术、电视技术,至于涉及到的数学、物理学等方面的基础知识就更多。

当今的图像处理理论大多是通信理论的推广,只是把通信中的一维问题推广到二维,以便于分析,在此基础上,逐步发展自己的理论体系。因此,图像处理技术与通信技术休戚相关。

在图像处理工程中的信息获取和显示技术主要源于电视技术,其中的摄像、显示、同步等各项技术是必不可少的。

计算机已是图像处理的常规工具,在图像处理中涉及到软件、硬件、网络、接口等多项技术,特别是并行处理技术在实时图像处理中显得十分重要。

图像处理技术的发展涉及越来越多的基础理论知识,雄厚的数理基础及相关的边缘学科知识对图像处理科学的发展有越来越大的影响。总之,图像处理科学是一项涉及多学科的综合性科学。

(3) 图像信息理论与通信理论密切相关

早在1948年,Shannon发表了"A Mathematical Theory of Communication"(通信中的数学理论)一文,奠定了信息论的基础。此后,信息理论渗透到了各个领域。图像信息论也属于信息论科学中的一个分支。从当今的理论发展看,我们可以说图像信息论是在通信理论研究的

基础上发展起来的。图像理论是把通信中的一维问题推广到二维空间上来研究的，也就是说，通信研究的是一维时间信息，图像研究的是二维空间信息；通信研究的是时间域和频率域的问题，图像理论研究的是空间域和空间频率域（或变换域）之间的关系；通信理论中认为：任何一个随时间变化的波形都是由许多频率不同、振幅不同的正弦波组合而成，图像理论认为：任何一幅平面图像是由许多频率、振幅不同的 X-Y 方向的空间频率波相叠加而成，高空间频率波决定图像的细节，低空间频率波决定图像的背景和动态范围。

总之，通信中的一维问题都可推广到二维，尽管有些理论尚不完全贴切，但对图像自身理论体系的形成有极大的借鉴意义。

1.4 数字图像处理的主要方法及主要内容

1.4.1 数字图像处理方法

数字图像处理方法大致可分为两大类，即：空域法和变换域法。

1. 空域法

这种方法是把图像看作是平面中各个像素组成的集合，然后直接对这一二维函数进行相应的处理。空域处理法主要有下面两大类：

(1) 邻域处理法

其中包括：梯度运算（Gradient Algorithm），拉普拉斯算子运算（Laplacian Operator），平滑算子运算（Smoothing Operator）和卷积运算（Convolution Algorithm）。

(2) 点处理法

灰度处理（grey processing），面积、周长、体积、重心运算等等。

2. 变换域法

数字图像处理的变换域处理方法是首先对图像进行正交变换，得到变换域系数阵列，然后再施行各种处理，处理后再反变换到空间域，得到处理结果。

这类处理包括：滤波、数据压缩、特征提取等处理。

1.4.2 数字图像处理的主要内容

完整的数字图像处理工程大体上可分为如下几个方面：图像信息的获取；图像信息的存储；图像信息的传送；图像信息处理；图像信息的输出和显示。

1. 图像信息的获取（Image information acquisition）

就数字图像处理而言，主要是把一幅图像转换成适合输入计算机或数字设备的数字信号，这一过程主要包括摄取图像、光电转换及数字化等几个步骤。通常图像获取的方法有如下几种：

(1) 电视摄像机（Video Camera）

这是目前使用最广泛的图像获取设备。早期主要有光电摄像管、超正析摄像管等。近年来，主要是采用 CCD 摄像设备。该设备有如下特点：

特点：设备小巧、速度快、成本低、灵敏度高。

缺点：灰度层次较差、非线性失真较大、有黑斑效应，在使用中需要校正。目前，CCD 摄像机在分辨率、灵敏度等方面已做到较高水平，如：1920×1035 或 1024×1024 的高分辨率的

CCD 摄像机已很成熟。

(2) 飞点扫描器(Flying point scanner)

这是一种以光源做扫描的图像获取设备。其特点是：精度较高、图像清晰、可透射成像亦可反射成像，但是其体积略显庞大。

(3) 扫描鼓

这是一种高精度的滚桶式的图像摄取设备。

特点：精度高、分辨率高，可以输入也可以输出。

缺点：价钱昂贵、速度低、维护要求高。多用于静止图像的输入、输出设备。

(4) 扫描仪

特点：精度和分辨率中等，600DPI 精度的扫描仪已常见。扫描仪的成本很低，近几年尤其降价显著，一般台式的已有不足一千元的产品。所以是当今应用最为广泛的图像信息获取设备。

缺点：速度较慢，非实时设备。

(5) 显微光密度计：精度较高，速度较低。

(6) 遥感中常用的图像获取设备已有多种，如：

光学摄影：摄像机、多光谱像机等。

红外摄影：红外辐射计、红外摄像仪、多通道红外扫描仪。

MSS：多光谱扫描仪。

微波：微波辐射计，侧视雷达、真实空孔径雷达、合成孔径雷达(SAR)。

合成孔径雷达是 20 世纪 50 年代发展起来的技术。它采用小天线通过直线飞行(长距离)合成一条很长的线阵天线，从而达到优良的横向方位的分辨率。目前的国际水平，在距雷达 50～100km 范围内，合成孔径雷达(SAR)的纵向和横向分辨率已达 1m×1m 以下。例如 $\lambda=3$cm，距雷达 50km 处的分辨率要达到 1m×1m 时所需的飞行直线合成孔径为

$$L = K\frac{\lambda R_0}{2\delta Y_a}$$

其中，$\lambda=3\text{cm}$，$K=1.35$，$R_0=50\text{km}$，$\delta Y_a=1$，$L=1012\text{m}$。

由此可以看出飞行距离很长。为使飞机能直线、恒速飞行要用到陀螺导航仪、GPS 定位系统等设备和技术加以保证。

2. 图像信息的存储(Image information storage)

图像信息的突出特点是数据量巨大。一般作档案存储主要采用磁带、磁盘或光盘。为解决海量存储问题主要研究数据压缩、图像格式及图像数据库技术等。

3. 图像信息的传送(Image information transmission)

图像信息的传送可分为系统内部传送与远距离传送。内部传送多采用 DMA 技术(Direct Memory Access)以解决速度问题，外部远距离传送主要解决占用带宽问题。目前，已有多种国际压缩标准来解决这一问题，图像通信网正在逐步建立。

4. 数字图像处理(Digital image processing)

目前，数字图像处理多采用计算机处理，因此，有时也称之为计算机图像处理 (Computer Image Processing)。数字图像处理概括地说主要包括如下几项内容：几何处理(Geometrical Processing)，算术处理(Arithmetic Processing)，图像增强(Image Enhancement)，图像复原(Image Restoration)，图像重建(Image Reconstruction)，图像编码(Image Encoding)，图像识

别(Image Recognition),图像理解(Image Understanding)。

(1) 几何处理

几何处理主要包括坐标变换,图像的放大、缩小、旋转、移动,多个图像配准,全景畸变校正,扭曲校正,周长、面积、体积计算等。

(2) 算术处理

算术处理主要对图像施以+、-、×、÷等运算,虽然该处理主要针对像素点的处理,但非常有用,如医学图像的减影处理就有显著的效果。

(3) 图像增强

图像增强处理主要是突出图像中感兴趣的信息,而减弱或去除不需要的信息,从而使有用信息得到加强,便于区分或解释。主要方法有直方图增强、伪彩色增强法(pseudo color)、灰度窗口等技术。

(4) 图像复原

图像复原处理的主要目的是去除干扰和模糊,恢复图像的本来面目。典型的例子如去噪就属于复原处理。图像噪声包括随机噪声和相干噪声,随机噪声干扰表现为麻点干扰,相干噪声表现为网纹干扰。去模糊也是复原处理的任务。这些模糊来自透镜散焦,相对运动,大气湍流,以及云层遮挡等。这些干扰可用维纳滤波、逆滤波、同态滤波等方法加以去除。

(5) 图像重建

几何处理、图像增强、图像复原都是从图像到图像的处理,即输入的原始数据是图像,处理后输出的也是图像,而重建处理则是从数据到图像的处理。也就是说输入的是某种数据,而处理结果得到的是图像。该处理的典型应用就是CT技术,CT技术发明于1972年,早期为X射线(X-ray)CT,后来发展的有ECT、超声CT、核磁共振(NMR)等。图像重建的主要算法有代数法、迭代法、傅里叶反投影法、卷积反投影法等,其中以卷积反投影法运用最为广泛,因为它的运算量小、速度快。值得注意的是三维重建算法发展得很快,而且由于与计算机图形学相结合,把多个二维图像合成三维图像,并加以光照模型和各种渲染技术,能生成各种具有强烈真实感及纯净的高质量图像。三维图形的主要算法有线框法、表面法、实体法、彩色分域法等等,这些算法在计算机图形学中都有详尽的介绍。三维重建技术也是当今颇为热门的虚拟现实和科学可视化技术的基础。

(6) 图像编码

图像编码研究属于信息论中信源编码范畴,其主要宗旨是利用图像信号的统计特性及人类视觉的生理学及心理学特性对图像信号进行高效编码,即研究数据压缩技术,以解决数据量大的矛盾。一般来说,图像编码的目的有三个:①减少数据存储量;②降低数据率以减少传输带宽;③压缩信息量,便于特征抽取,为识别作准备。就编码而言,Kunt提出第一代、第二代编码的概念。Kunt把1948—1988年40年中研究的以去除冗余为基础的编码方法称为第一代编码。如:PCM、DPCM、ΔM、亚取样编码法;变换编码中的DFT、DCT、Walsh-Hadamard变换等方法以及以此为基础的混合编码法均属于经典的第一代编码法。而第二代编码方法多是20世纪80年代以后提出的新的编码方法,如金字塔编码法、Fractal编码、基于神经元网络的编码方法、小波变换编码法、模型基编码法等。现代编码法的特点是:①充分考虑人的视觉特性;②恰当地考虑对图像信号的分解与表述;③采用图像的合成与识别方案压缩数据率。

图像编码应是经典的研究课题,60多年的研究已有多种成熟的方法得到应用。随着多媒体技术的发展,已有若干编码标准由ITU-T制定出来,如JPEG、H.261、H.263、MPEG1、

MPEG2、MPEG4、MPEG7、JBIG(Joint Bi-level Image Coding Expert Group,二值图像压缩)等。相信在未来会有更多、更有效的编码方法问世,以满足多媒体信息处理及通信的需要。

(7) 模式识别

模式识别是数字图像处理的又一研究领域。当今,模式识别方法大致有三种,即:统计识别法;句法结构模式识别法;模糊识别法。

统计识别法侧重于特征,句法结构识别侧重于结构和基元,模糊识别法是把模糊数学的一些概念和理论用于识别处理。在模糊识别处理中充分考虑人的主观概率,同时也考虑了人的非逻辑思维方法及人的生理、心理反映,这一独特性的识别方法目前正处于研究阶段,方法尚未成熟。

(8) 图像理解

图像理解是由模式识别发展起来的方法。该处理输入的是图像,输出的是一种描述。这种描述并不仅是单纯的用符号作出详细的描绘,而且要利用客观世界的知识使计算机进行联想、思考及推论,从而理解图像所表现的内容。图像理解有时也叫景物理解。在这一领域还有相当多的问题需要进行深入研究。

以上所述的8项处理任务是图像处理所涉及到的主要内容。总的说来,经多年的发展,图像处理经历了从静止图像到活动图像;从单色图像到彩色图像;从客观图像到主观图像;从二维图像到三维图像的发展历程。特别是与计算机图形学的结合已能产生高度逼真、非常纯净、更有创造性的图像。由此派生出来的虚拟现实技术的发展或许将从根本上改变我们的学习、生产和生活方式。

5. 图像的输出与显示

图像处理的最终目的是为人或机器提供一幅更便于解译和识别的图像。因此,图像输出也是图像处理的重要内容之一。图像的输出有二种,一种是硬拷贝,另一种是软拷贝。其分辨率随着科学技术的发展从256×256、512×512、1024×1024,至今已有2048×2048的高分辨率的显示设备问世。通常的硬拷贝方法有照相、激光拷贝、彩色喷墨打印等几种方法。软拷贝方法有以下几种:

(1) CRT显示(Cathode Ray Tube)

自20世纪60年代以来,在显示技术中,CRT几乎独霸天下。目前,彩色显像管(CPT)和彩色显示管(CDT)技术已相当成熟。90年代后期平板显示器件才相继问世。但专家们预测在未来10年内CRT仍是图像显示的主流。CRT显示质量好、亮度高、电子束寻址方式简单、制造成本低等都是CRT的显著优点。尤其采用微形滤光条(Microfilter)工艺,加之动态聚焦技术的出现,使得CRT在对比度、色纯及光点大小方面都得到了改进。目前,分辨率为1280×1024、行频64kHz、点频110MHz的CRT已很普遍,高分辨率的可达到1920×1035,行频达80kHz,视频带宽达140MHz。进一步提高分辨率的主要困难在于显像管的制造和刷新存储器的速度。我国生产的CPT年产达1800万个,而CDT却很少,国内只有两个厂家,年产只有35万个。从市场占有率来看CRT亦是主流。

(2) 液晶显示器(LCD)

液晶的发现已有100多年的历史,但真正用于显示技术的历史不到30年。尽管有人认为LCD要想取代CRT至少还需15年左右的时间,但其发展势头之大,发展速度之快却令人刮目相看。LCD的突出性能是极吸引人的,它的缺点正在逐步被克服。最近的最新产品已大有改善,如Sharp公司推出的彩色非晶硅TFT-LCD产品,屏幕尺寸21英寸,分辨率640×480,像

素数 921600 点，彩色数 1670 万种。富士通推出的 10.4 英寸显示器的视角可达 120°。

(3) 场致发光显示器(FED)

场致发光平面显示器有多种。总体来说，从技术上看还不能与 CRT 和 LCD 相竞争。但等离子显示器件的性能优于 LCD。其本身视角可达 160°，结构工艺简单，目前是有力的竞争者。1994 年已有 40 英寸的壁挂式 AD-PDP 显示器展出。至于彩色荧光显示目前只能用于字符显示。

场致发光显示(FED)具有光明的前途。FED 是最新发展起来的彩色平板显示器件。SID95' 会议上展出的产品 6 英寸方形的全色 FED 分辨率已达到 512×512。但目前要解决的是大面积的 FED 需要改善发光的均匀性和提高低压荧光粉的发光效率，在实用化中，封接、排气、真空维持等工艺尚有困难，这些问题解决后 FED 大有前途。

1.5 数字图像处理的硬件设备

一般的数字图像处理系统由下图 1-1 所示。

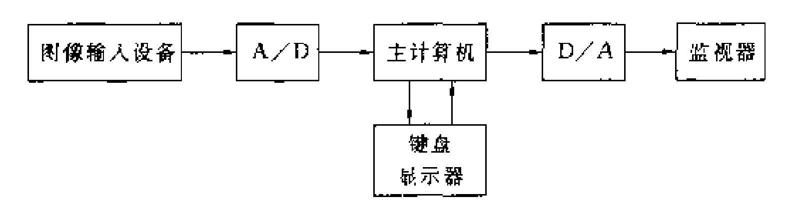


图 1-1 数字图像处理系统

早期的数字图像处理系统为提高处理速度，增加容量，都采用大型机。大型机的造价高，浪费大。后来较普遍的是发展小型机为主的系统。具有代表性的是美国的 I^2S 系统，加拿大的 Depex系统。这种系统的主机均以 VAX/750，VAX/785 为主。现在的图像处理系统向两个方向发展，一个方向是微型图像处理系统，主机为 PC 机，配以图像卡及显示设备就构成了最基本的微型图像处理系统。目前，国产的有 CA540、Vp32、FGCT11010N8、CA-CPE-1000、CA-CPE-3000 等图像板研制成功并已商品化。多媒体系统中常用的 Video Blaster 也是一种较普遍的图像卡。此外，大多数工作站也都有图像处理功能，也可以看作是微型图像处理系统，但一般工作站大都以网络性能和图形功能见长。

微型系统成本低、设备紧凑、应用灵活、便于推广。特别是微型计算机的性能逐年提高，使得微型图像处理系统的性能也不断升级，加之软件配置丰富，使其更具实用意义。

图像处理系统的另一方向就是向大型机方向发展，以解决大数据量与处理能力之间的矛盾。

当前要从根本上解决处理能力、速度与数据量巨大的问题，还应该发展阵列机和并行处理技术，早期的几种有代表性的阵列处理器有如下几种：

(1) AP-120B

AP-120B 是美国 Floating Point 公司研制的并行处理器，采用流水线结构，可用于空间滤波及 FFT 等处理。它有专用的浮点乘法器和加法器。对 512×512 的图像作 FFT，用 1.6s，比当时的大型机快几百倍。

(2) FP 处理器(Flexible Processor)

该处理器进行 8 位的乘法运算只要 250ns。该处理器广泛用于匹配计算、相关计算、图像

空间坐标转换、灰度校正等。

(3) 完全并行处理器

英国 London 大学曾研制 CLIP-4(Cellule Logic Image Processor),该处理器由 96×96 的基本细胞单元组成,由一台 PDP-11 做控制,处理速度可达 10 亿次/秒。

(4) 局部并行处理器

完全并行处理器的速度快,但运算电路数量多,而局部并行处理器却有一定的优点,因此应用也较广。

① Pattern Processing

日本东京大阪大学研制的并行处理器,可做 5×5 窗口的门限处理,微分处理、细化处理等。对 64×64 的图像用 3×3 窗口作微分处理需要 12ms,细化处理要 15ms。

② Image Processor

由日本日立中央制作所研制,该处理器可做微分处理、平均化、数据变换、直方图修改等,256×256 大小的图像要 0.2~1.2s。

③ PPP 系统(Parallel Pattern Processor)

PPP 系统是日本东芝公司研制的。该系统采用流水线与并行处理方式,并行处理器有 8 个加法器 8 个乘法器,可做滤波处理,变换处理,1μs 可做 64 次乘法和 64 次加法,同时还有 128 次存取操作。

总之,并行处理技术不仅是计算机科学中的重要研究对象,在实时图像处理中也是不可缺少的关键技术。

1.6 数字图像处理的应用

数字图像处理的应用越来越广,已经渗透到工程、工业、医疗保健、航空航天、军事、科研、安全保卫等各个方面,在国计民生及国民经济中发挥越来越大的作用。

具体应用领域可粗略概括为表 1-1。

表 1-1 图像处理的应用领域

学　科	应用内容
物理、化学	结晶分析、谱分析
生物、医学	细胞分析、染色体分类、血球分类、X 射线照片分析、CT
环境保护	水质及大气污染调查
地质	资源勘探、地图绘制、GIS
农林	植被分布调查、农作物估产
海洋	鱼群探查、海洋污染监测
水利	河流分布、水利及水害调查
气象	云图分析等
通信	传真、电视、多媒体通信
工业、交通	工业探伤、铁路选线、机器人、产品质量监测
经济	电子商务、身份认证、防伪
军事	军事侦察、导弹制导、电子沙盘、军事训练等
法律	指纹识别等

应用的典型例子有：

(1) 遥感

在遥感的发展和大事纪中，我们可以看到大量的与图像处理密切相关的技术。从世界上出现第一幅照片(1839 年)、意大利人乘飞机拍摄了第一张照片(1909 年)、苏联(1957 年)及美国(1958 年)发射第一颗人造地球卫星等都为遥感技术的发展奠定了坚实的基础。1962 年国际上正式使用遥感一词(Remote Sensing)。此后，美国相继发射多颗陆地资源探测卫星(1972 年，LANDSAT-Ⅰ——四个波段，地面分辨率 59m×79m；1975 年，LANDSAT-Ⅱ；1978 年，LANDSAT-Ⅲ，分辨率 40m×40m；1982 年，LANDSAT-Ⅳ，分辨率 30m×30m，在这颗卫星上配置了 GPS 系统(Global Positioning System)，定位精度在地心坐标系中为±10m。)

遥感图像处理的用处越来越大，效率及分辨率也越来越高。如：土地测绘、资源调查、气象监测，环境污染监测、农作物估产、军事侦察等。当前，在遥感图像处理中主要解决数据量大和处理速度慢的矛盾。

(2) 医学应用

图像处理在医学界的应用非常广泛，无论是在临床诊断还是病理研究都大量采用图像处理技术。它的直观、无创伤、安全方便的优点受到普遍的欢迎与接受。其主要应用可举出众多的例子，如 X 射线照片的分析、血球计数与染色体分类等。目前广泛应用于临床诊断和治疗的各种成像技术，如超声波诊断等都用到图像处理技术。有人认为计算机图像处理在医学上应用最成功的例子就是 X 射线 CT(X-ray Computed Tomography)。1968—1972 年英国的 EMI 公司的 Hounsfeld 研制了头部 CT，1975 年又研制了全身 CT。70 年代下半叶美、日、法、荷兰相继生产 CT。其中主要研制者 Hounsfeld(英)和 Commack(美)获得了 1979 年的诺贝尔生理医学奖。这足以说明 CT 的发明与研究对人类的贡献之大、影响之深。类似的设备目前已有多种，如核磁共振 CT(NMRI，Nuclear Magnetic Resonance Imaging)，电阻抗断层图像技术(EIT，Electrical Impedance Tomography)和阻抗成像(Impedance Imaging)，这是一种利用人体组织的电特性(阻抗、导纳、介电常数)形成人体内部图像的技术。由于不同组织和器官具有不同的电特性。因此，这些电特性包含了解剖学信息。更重要的是人体组织的电特性随器管功能的状态而变化，因此，EIT 可望绘出反应与人体病理和生理状态相应功能的图像。目前，EIT 已发展了一些相应的算法(如图像重建算法)，并在临床应用中也正在探索(如：神经中枢系统、呼吸系统、心血管系统、消化系统)。当前的主要问题是分辨能力差，原因是人射电流进人人体组织后呈三维分布发散，因此，指向性不强，并且电流在人体组织中的分布规律复杂，未知因素多。虽然 EIT 分辨率不高，但是生物阻抗技术提取的组织和器官的电特性信息对血液、气体、体液和不同组织成份有独特的鉴别能力，对血液的流动分布，肺内的气血交换，体液含量与流动等非常敏感，以此为基础，可进行心、脑、肺及相关循环系统的功能评价及血液动力学与流变学的研究。该技术对肺癌的早期发现显示出很大优越性，这一点是现有的其他成像技术无法比拟的。

(3) 图像处理技术在通信中的应用

图像通信如按业务性能划分可分为电视广播(点对面通信)、传真、可视电话(点对点通信)、会议电视(点电对多点)、图文电视、可视图文以及电缆电视等。如按图像变化性质分，可分为静止图像和活动图像通信。

从历史上看，早在 1865 年就在法国试验成功传真通信(巴黎至里昂)，但后来由于技术及经济原因发展一直非常缓慢。70 年代后，图像通信逐渐成为人们生活中常用的通信方式，随着大规模集成电路的发展，使得图像通信中所需的关键技术逐步得到解决，推动了图像通信的发

展。1980年CCITT为三类传真机和公共电话交换网上工作的数字传真建立了国际标准，1984年CCITT提出了ISDN的建议，以及当今基于IP的多媒体通信都意味着非话业务通信方式已在通信中占有重要位置。图像通信主要有如下一些内容：

① 电视广播：单色电视广播1925年在美国实现。1936年BBC开始电视广播。目前出现的彩色电视有三种制式，即NTSC（美国、日本等）、PAL（中国、西欧、非洲等）和SECAM（法国、俄罗斯等）。

② 可视电话和会议电视：1964年美国国际博览会展出了Picture-phone MOD-I可视电话系统，带宽为1MHz。目前的可视电话/会议电视均采用数字压缩技术，也出现了相应的国际标准。如：图像编码标准H. 261、H. 263等，会议电视的H. 230标准，在专用通信网中用PCM一次群传输，速率为2048kb/s。桌面型系统遵循H. 323标准。

③ 传真：是把文字、图表、照片等静止图像通过光电扫描的方式变成电信号加以传送的设备。1980年CCITT为三类传真机和公共电话交换网上工作的数字传真建立了国际标准，即：一类机——不压缩，4线/毫米，A4文件传6分钟；二类机——采用频带压缩技术（残留边带传输），4线/毫米，传A4文件需3分钟；三类机——在传送前采用去冗余技术，在电话线上以1分钟传A4文件；四类机——在三类机的基础上发展的采用去冗余技术的传真设备，采用去冗余、纠错码技术在公用数据网上使用的设备，加Modem也可以在公用电话网上使用。经过多年发展，传真技术不断进步，现在已有仅数秒钟就可传送一幅A4文件的传真机，分辨率高达16点/毫米。

④ 图文电视和可视图文：图文电视（Teletext）和可视图文（Videotex）是提供可视图形文字信息的通信方式。图文电视是单向传送信息，它是在电视信号消隐期发送图文信息，用户可用电视机和专用终端收看该信息；可视图文是双向工作方式，用户可用电话向信息中心提出服务内容或从数据库中选择信息。

⑤ 电缆电视（CATV）：是通过电缆或光缆传送的电视节目。第一个电缆电视系统于1949年安装在美国，采用光缆实现的CATV是1977年后的事情。

（4）工业生产的质量控制

在生产线中对生产的产品及部件进行无损检测也是图像处理技术的一个广泛的应用领域。如食品包装出厂前的质量检查，浮法玻璃生产线上对玻璃质量的监控和筛选，甚至在工件尺寸测量方面也可以采用图像处理的方法加以自动实现。另外，铁谱分析也是一个典型的应用。

（5）安全保障、公安等方面的应用

该领域可采用模式识别等方法实现监控、指纹档案、案件侦破、交通管理等。

（6）教学和科研领域

教学及科研领域中也将大量应用图像处理技术。如科学可视化技术，远程培训及教学也将大量使用图像处理技术的成果。

（7）电子商务

当前呼声甚高的电子商务中，图像处理技术也大有可为。如：身份认证、产品防伪、水印技术等。

总之，图像处理技术应用还可列出相当多的领域，它在国家安全、经济发展、日常生活充当越来越重要的角色，对国计民生的作用不可低估。

1.7 数字图像处理领域的发展动向

自 60 年代第三代数字计算机问世以后，数字图像处理技术出现了空前的发展，其形势可谓方兴未艾。

在该领域中需进一步研究的问题，不外乎如下五个方面：

1）在进一步提高精度的同时着重解决处理速度问题。如在航天遥感、气象云图处理方面，巨大的数据量和处理速度仍然是主要矛盾之一。

2）加强软件研究、开发新的处理方法，特别要注意移植和借鉴其他学科的技术和研究成果，创造新的处理方法。

3）加强边缘学科的研究工作，促进图像处理技术的发展。如：人的视觉特性、心理学特性等的研究如果有所突破，将对图像处理技术的发展起到极大的促进作用。

4）加强理论研究，逐步形成图像处理科学自身的理论体系。

5）时刻注意图像处理领域的标准化问题。图像的信息量大、数据量大，因而图像信息的建库、检索和交流是一个极严重的问题。就现有的情况看，软件、硬件种类繁多，交流和使用极为不便，这成了资源共享的严重障碍。应及早建立图像信息库，统一存放格式，建立标准子程序，统一检索方法。

图像处理技术未来发展大致可归纳为如下四点：

1）图像处理的发展将向着高速、高分辨率、立体化、多媒体化、智能化和标准化方向发展。围绕着 HDTV（高清晰度电视）的研制将开展实时图像处理的理论及技术研究。其中包括：

① 提高硬件速度

不仅要提高计算机的速度，而且 A/D 和 D/A 的速度都要实时化。提高计算机速度的途径有三个：其一，对于复杂指令计算机（CISC）的处理速度在逐年提高，如 Intel 公司的芯片，486 有一百多万个元件，速度不到 10Mips（Million instruction per second），到 2000 年生产的 Micro-2000 每片可集成一亿个元件，速度达 200Mi/s；其二、对于精简指令计算机（RISC）将给予极大的重视，20 世纪 90 年代为 10Mi/s，1995 年达到 800Mi/s，利用 ECL 电路速度还会增加。有人预言到 2000 年速度会超过 2000Mi/s；第三个提高图像处理速度的途径是研究并行处理器，利用多个 CPU 并联，使软硬件一体化。在日本有人估计到 2010 年将有 10^8～10^{12}个处理器并行，大体与人脑的神经元数量相当，速度可达到 GFlops（1G＝1000M）甚至 TFlops（1T＝1000G）。

② 提高分辨率

主要提高采集分辨率和显示分辨率。20 世纪 90 年代达到 2048×2048。提高分辨率的主要困难在于显像管的制造和图像、图形刷新存取速度。20 世纪 80 年代法国的 SPOT 卫星的分辨率为 10m×10m，由此可见，分辨率的提高速度是十分惊人的。

③ 立体化

图像是二维信息。随着技术的发展，三维图像处理将会更受重视，因为它的信息量更大，特别是随着计算机图形学及虚拟现实技术的发展，立体图像处理技术将会得到广泛应用。

④ 多媒体化

20 世纪 90 年代出现的多媒体技术在计算机界掀起了一股热潮，现在这一词汇可以说到处可见，实际上多媒体的关键技术就体现在图像数据压缩上。

目前有关图像压缩的国际标准已有多个，而且还在继续发展。如：

· JPEG (Joint Photography Expert Group):1991 年 3 月提出的 ISO CD 10918 号建议。

· H.261 建议:用于可视电话/会议电视的压缩标准。1998 年提出 $P\times 64$Kb/s, P 值可取 1～30,采用混合编码法,即采用 DCT 变换、运动补偿 DPCM、Huffman 编码技术。1984 年制定了会议电视的 H.120 建议。

· MPEG1 标准

1992 年通过 ISO CD 11172 号建议,它包括三部分,即 MPEG 视频、MPEG 音频、MPEG 系统,速率为 1.5Mb/s。目前的 MPEG4 是为实现多媒体通信的国际标准,主要把 MPEG 的典型特点与现有的或将来预期会出现的特征结合起来的新的编码标准。

总之多媒体技术的进一步发展是使计算机朝着人类接收和处理信息的最自然的方式发展。

⑤ 智能化

力争使计算机的识别和理解能够按人的认识和思维方式工作,考虑主观概率,非逻辑思维,正如微软提出的要研制能听会说的计算机那样实现多功能的人机交互。

⑥ 标准化

在图像处理技术整体看尚没有国际标准,今后应给予极大的关注。

2) 图像、图形相结合朝着三维成像或多维成像的方向发展。

3) 硬件芯片研究。

目前,结合多媒体技术的研究,硬件芯片越来越多。如 Thomson 公司 ST13220 采用 Systolic 结构,作运动预测器。INMOS 公司的 IMS-A121,采用流水线结构, C-Cube 公司 CL-550 把 JPEG 做到一个芯片上,更便于推广应用等。总之把图像处理的众多功能固化在芯片上将会有更加广阔的应用领域。

4) 新理论与新算法研究。

在图像处理领域近年来引入了一些新的理论并提出了一些新的算法,如:Wavelet、Fractal、Morphology、遗传算法、神经网络等。其中 Fractal 广泛用于图像处理、图形处理、纹理分析,同时还可以用于数学、物理、生物、神经和音乐等方面,有人认为 Fractal 把杂乱无章、随意性很强的事物能用数学方法加以规范和描述,它在分析和描绘自然现象上具有独到之处,这些理论在未来图像处理理论与技术上的作用应给予充分的注意,并积极地加以研究。

图像处理特别是数字图像处理科学经初创期、发展期、普及期及广泛应用几个阶段,如今已是各个学科竞相研究并在各个领域广泛应用的一门科学。今天,随着科技事业的进步以及人类需求的多样化发展,多学科的交叉、融合已是现代科学发展的突出特色和必然途径,而图像处理科学又是一门与国计民生紧密相连的一门应用科学,它的发展与应用与我国的现代化建设联系之密切、影响之深远是不可估量的。图像处理科学无论是在理论上还是实践上都存在着巨大的潜力。

思 考 题

1. 图像处理的主要方法分几大类?
2. 图像处理工程包括哪几项内容?
3. 数字图像处理的主要内容是什么?

4. 什么是软拷贝？什么是硬拷贝？
5. 图像信息获取设备有哪几种？其优缺点是什么？
6. 图像显示的主要方法有几种？
7. 数字图像处理有哪些应用？
8. 数字图像处理的发展方向是什么？

第2章　图像、图像系统与视觉系统

在数字图像处理技术中，处理方案的选择和设计与信源和信宿的特性密切相关。所谓信源就是处理前或处理后的图像，而信宿就是处理前后图像信息的接收者——人的视觉系统。因此了解图像的特点及人的视觉系统的特性是恰当地选择处理图像的方法以便从中获取最大的信息量所必备的先验知识。

本章将对图像的客观性质、图像处理系统的外围设备及人眼的视觉特性进行一些必要的讨论。

2.1　图像

2.1.1　有关光学的预备知识

在讨论图像与视觉系统的过程中，必然要涉及到光学的有关知识。因此，我们首先介绍一下有关光学的术语及计量单位，以便为后续内容的展开作一点铺垫。

光学理论中用到的主要术语及计量单位如下：

(1) 发光强度 I (intensity)

光源发光的功率称为发光强度。其单位主要有如下两种：

烛光(Candle Power，c)　1c 是指标准蜡烛发出的光。标准蜡烛是用鲸脑油制成，重 1/6 磅，燃烧率为 120 格令(1 格令＝0.0648 克)的蜡烛。

坎德拉(Candle，cd)　1cd 就是“全辐射体”加温到铂的熔点(2024K)时从 1cm^2 表面面积上发出的光的 1/60。所谓“全辐射体”就是某一物质加热到某一温度时，它发出的能量分布在整个可见光范围内。理论上的全辐射体就是一个完全黑体，当冷却后，它将吸收所有入射到它上面的光。

在实用中可以认为 1c＝1cd。

(2) 光通量 Φ

光通量是每秒钟内光流量的度量，其单位是流明(lm)。

流明是指与 1cd 的光源相距的单位距离，与入射光相垂直的单位面积上每秒钟流经的光流量叫 1 lm。

(3) 照度 E (illumination)

入射到某表面的光通量密度称为该表面的照度。用每单位面积的流明数来表示。主要单位有如下几种：

公制单位：

勒克司 (lx)　$1\text{lx}=1\ \text{lm}/\text{cm}^2$

辐透 (phot)　$1\text{phot}=1\ \text{lm}/\text{cm}^2$

英制单位：

英尺·烛光　1 英尺·烛光=1 lm/ft^2

一般换算关系如下：

1 英尺·烛光 = 1 流明/英尺^2 = 10.76 lm/m^2 = 10.76 lx

(4) 反射

反射系数　$\rho = \frac{\text{表面反射的流明数}}{\text{入射到该表面的流明数}}$

(5) 透射

透射系数　$\tau = \frac{\text{物质透射的流明数}}{\text{入射到该物质的流明数}}$

(6) 亮度

这个概念用来说明物体表面发光的量度，光可以由一个面光源直接辐射出来，也可以由入射光照射下的某表面反射出来。亮度对其两者均适用。

亮度的衡量有各种不同的单位，其中主要有 A、B 两组。A 组是以每单位面积上的发光强度来表示的；B 组是以每单位面积上发出的光通量来表示的。当然这两组单位是可以换算的。

A 组：使用坎德拉为单位。

尼特　1 尼特=1 坎德拉/米^2(cd/m^2)

熙提　1 熙提=1 坎德拉/厘米^2(cd/cm^2)

　　1 熙提=10^4 尼特

B 组：使用流明数为单位。

亚熙提　1 亚熙提=1 流明/米^2(lm/m^2)

朗伯 L　1 朗伯=1 流明/厘米^2(lm/cm^2)

　　1 朗伯=10^4 亚熙提

相应的英制单位如下：

A 组

1 坎德拉/英尺^2(cd/ft^2)

1 坎德拉/英寸^2(cd/in^2)

B 组

1 英尺·朗伯=1 流明/英尺^2(lm/ft^2)

换算关系：

1 尼特=3.14 亚熙提

1 熙提=3.14 朗伯

1 坎德拉/英尺^2=10.76 尼特

1 英尺·朗伯=10.76 亚熙提

1 坎德拉/英尺^2=3.14 英尺·朗伯

由于光学单位名目繁多，往往容易引起混乱，所以又提出 SI 单位(国际单位)，见表 2-1。

表 2-1　SI 单位

被测量	SI 单位	缩写	被测量	SI 单位	缩写
发光强度(I)	坎德拉	cd	照度(E)	勒克司	lx
光通量(Φ)	流明	lm	亮度(L)	坎德拉/米^2	cd/m^2

2.1.2 图像的概念

"图像"一词在汉语中很难给出一个明确的定义。当我们打开英语词典时可以找到三个与图像有关的词,那就是 picture、image 和 pattern。一般英文词典对这三个词是这样注释的,picture——画、图画、图像、图片、电影等等;image——像、图象、景象、映像、影像、反射、映射等等;pattern——模型、式样、样本、图案、花样、图、图形等等。从这三个词的注释中大致可做如下区分,其中 picture 是指与照片等相似的用手工描绘的人物或景物,其中侧重于手工描绘的一类"画"。image 是指用镜头等科技手段得到的视觉形象。一般来讲可定义为"以某一技术手段被再现于二维画面上的视觉信息"。通俗地说就是指那些用技术手段把目标(object)原封不动的一模一样地再现的图像。它包含用计算机等机器产生的景物。而 pattern 指的是图形,在拉丁语中指裁衣服的纸样。因此,它主要是指图案、曲线、图形。综上所述我们所说的图像处理应是 Image Processing。这里,我们要处理的主要是属于照片、复印图、电视、传真、计算机显示的一类图像。

当用数学方法描述图像信息时,通常着重于考虑它的点的性质。例如一幅图像可以被看成是空间各个坐标点上强度的集合。它的最普遍的数学表达式为

$$I = f(x,y,z,\lambda,t) \tag{2-1}$$

其中(x,y,z)是空间坐标,λ是波长,t是时间,I是图像的强度。这样一个表达式可以代表一幅活动的、彩色的、立体图像。

当我们研究的是静止图像(Still Image)时,则上式与时间t无关,当研究的是单色图像时,显然与波长λ无关,对于平面图像来说则与坐标z无关。因此,对于静止的、平面的、单色的图像来说其数学表达式可简化为

$$I = f(x,y) \tag{2-2}$$

上式说明,一幅平面图像可以用二维亮度函数来表示。因为光也是能量的一种表现形式,所以,

$$0 < f(x,y) < \infty \tag{2-3}$$

人们所感受到的图像一般都是由物体反射的光组成的。$f(x,y)$可以看成由两个分量组成,一个是我们所看到的景物上的入射光量,另一分量是景物中被物体反射的光量,它们可分别被称为照射分量和反射分量。如果用$i(x,y)$表示照射分量,用$r(x,y)$表示反射分量,那么,

$$I = f(x,y) = i(x,y) \cdot r(x,y) \tag{2-4}$$

式中:

$$0 < i(x,y) < \infty \tag{2-5}$$

$$0 < r(x,y) < 1 \tag{2-6}$$

其中(2-6)式表示全吸收情况为0,全反射的情况为1。这里$i(x,y)$由光源的性质来确定,而$r(x,y)$则取决于景物中的物体。

$i(x,y)$的单位用照度来度量,即 lm/m^2 或 lx。

下面我们开列出一些$i(x,y)$的典型值,以便为读者建立一点初步的感性认识。

例如:

晴朗的日子,太阳在地球表面造成的照度为 9000 英尺·烛光,也就是 96840 勒克司或者 96840 流明/米^2。

当天空有云时,太阳在地球表面造成的照度为 1000 英尺·烛光或 10760 流明/米^2,也就是

10760 勒克司。

晴天的夜晚而且是满月的情况下，地球表面的照度为 0.01 英尺·烛光=0.1076 勒克司

一般房间照明充分的室内照度大约为 100 英尺·烛光=1076 勒克司。

$r(x,y)$是反射系数,其典型物质的典型值如下：

黑天鹅绒	0.01；
不锈钢	0.65；
白色墙	0.80；
镀银金属	0.90；
白雪	0.98。

在数字图像处理中经常用到监视器或电视机。自然景物映射到摄像管靶面的光的强弱取决于景物上反射出来的光通量。一般黑白或彩色电视机的屏幕亮度约为 80～120 坎德拉/米^2 或 80～120 尼特。

2.1.3 图像信息的分类

图像信息的种类是多种多样的,但是要想进行明确的分类也并非容易。这里我们就信息处理中常见的图像信息进行一下简单的分类。

概括起来,图像信息大致可分成三类,即符号信息、景物信息和情绪信息。下面就这三种图像信息的特点做一讨论。

1. 符号图像信息

在这类信息中，一般是用文字、符号、图形等表示的具体的或抽象的事物。例如文字,利用文字可组成文章,在某种意义上也可以看成是用二值图像的形式携带这篇文章的寓意。最有代表意义的符号图像信息是电路图、机械图、建筑图等,它们都是用二值图像的形式向人们提供信息的。因为符号信息是以某一规则排列的记号,因此,在传送及处理中只要能表达清楚就可以了,它允许有较大的压缩。

2. 景物信息

这是一种能给人以主观感觉但并不取决于人本身的客观场景信息。一般来讲,它包含丰富的内容,所含的信息量也较多。如：由铁路调车场控制中心的工业电视上看到的图像信息,可从中得到有关车辆编组调度情况、调车员的工作情景及天气情况等等。情景画面的内容一般比较复杂,在传输和处理中做到较大的压缩比较困难。在人机识别中需要较大的信息量。但在事先设定某种条件的情况下,是有可能在任何情况下保证正确判断的。

3. 情绪信息

这是一类依赖于受信者的图像信息,它不仅能给人以直观感觉,而且能以其特殊的艺术内容刺激人的感官,使受信者"触景生情"引起感情上的波动和情绪上的共鸣。因此,它包含有更多的信息量。例如，当我们在春光明媚的季节漫步郊外时,看到春回大地、万物更新的一派生机勃勃的场面时,必然为这盎然的春意所感动,自然会产生一种难以名状的喜悦之感。当我们看到雄伟壮丽的万里长城,自然会联想到古代劳动人们的勤劳智慧,一种庄严的自豪感便由然而生。当我们在影片中看到雷电交加,周围一片漆黑的场面时,必然感到恐惧。当看到天色灰暗,霪雨绵绵的场景则会有无限的压抑之感。凡此种种都说明了这类图像信息不仅取决于图像本身的内容而且还与受信者的经历、文化修养、年龄、嗜好以及此时此刻的心境情绪有密切关系。换句话说,对于同一幅图像来说,它对受信者产生的效果却是有差异的。因此,对于这类图

像不仅无法考虑其概率模型，而且用仙农(Shannon)理论明确其信息量也是极其困难的。

以上是从图像所携带的信息的种类出发进行简单分类的。当然还可以从其他角度出发进行分类。如把图像分成静止图像和活动图像、单色图像和彩色图像等。在 Matre 所著《图像处理》一书中曾谈到 Levialdi 曾对图像分类做了较彻底的研究。其中甚至考虑了许多不寻常的图像。他把图像分为四种范畴，即：①明显图像　它包括幻觉、思虑图像、无意识图像等；②前后连贯的图像　如梦境图像、精神心里的幻觉等；③容存的图像　如记忆图像、眼内构成的图像等；④与感觉相影响的图像　如错觉、感觉失真、联想、模仿的图像等。

在我们数字图像处理中所涉及到的是一些最普通类型的图像，它们的突出特点是都具有特殊的统计特性，并且有专门的应用。从这个基点出发可做如下较明快的分类。

① TV 型的自然风景　这是常见的图片，如肖像、风景画、街道、建筑物照片等。

② 空间摄影照片和地球资源探测图片　这类图片的特点是往往没有适宜的方向，构图不十分明显，除了海岸线外，没有可区别的形状。

③ 电子显微镜照片和标准的显微镜照片　这是一类在冶金学、生物学、医学以及石油探测等都很感兴趣的一类照片。

④ 文本　这是指一类打印或手写的记号图像。

⑤ 图样　它们通常就是简单地由线段和图形构成的单色二值图像。

⑥ 专用图像　如：X 射线照片、微波照片、红外热像或超声波图像等。这些图像各有特点，与在可见光下得到的图像有所不同。

总之，我们的物质世界是一个无处不充满图像的世界，对这么多的图像进行分类无疑是十分困难的事。在本书中所提到的一些图像只是极少数的有代表性而又实用的图像。这些图像经过研究大部分可以找到较为近似的模型和规律，这对方便处理和深入研究来说无疑都是十分有利的。

2.1.4 图像的统计特性

在图像的统计特性表征中，认为图像信号是一个随机信号。对于一个随机信号的数学描述则是振幅或相位的分布函数、概率密度函数以及一系列的相关矩、中心矩、功率谱等等。利用这些参数来表征图像的特性，建立图像信息的数学模型，以便对图像信息进行有效的分析及处理。

1. 图像的振幅分布特性

图像信号的振幅分布特性由振幅分布函数或振幅分布密度函数来表示。振幅分布函数由下式表示：

$$F(z) = P\{g(x,y) < z\} \tag{2-7}$$

式中，$F(z)$是振幅分布函数，$g(x,y)$是二维图像信号，P 代表概率。由式(2-7)可知，所谓图像信号振幅分布函数就是图像信号 $g(x,y)$之值小于某一给定值 z 的概率。$g(x,y)$的值落在 z_1、z_2 之间的概率由下式表示：

$$F(z_2) - F(z_1) = P\{z_1 \leqslant g(x,y) < z_2\} \tag{2-8}$$

振幅分布密度函数 $f(z)$可由 $F(z)$的导数得到，即：

$$f(z) = \frac{\mathrm{d}}{\mathrm{d}z}F(z) \tag{2-9}$$

或者由式(2-10)来表示：

$$f(z)=\lim_{\Delta z\to 0}\frac{1}{\Delta z}P\{z\leqslant g(x,y)<z+\Delta z\} \tag{2-10}$$

图像信号的振幅分布函数及分布密度函数可由式(2-7)及式(2-10)测定。

日本 NHK 技研所的千叶、安东曾对三种图像的振幅分布特性作了测量。其中有戏剧场面摄影作品，电影镜头画面等。其测试曲线如图 2-1 所示。

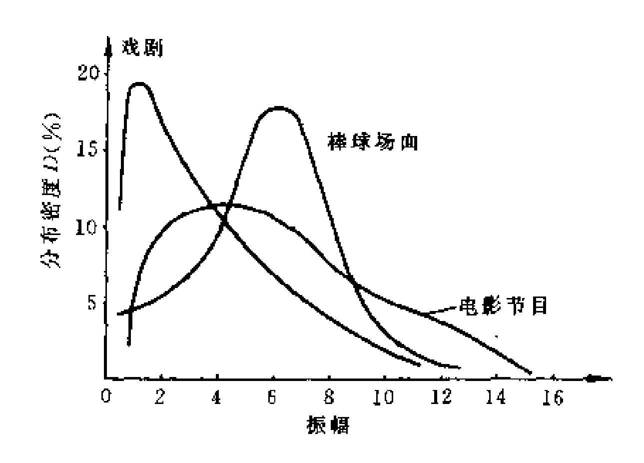


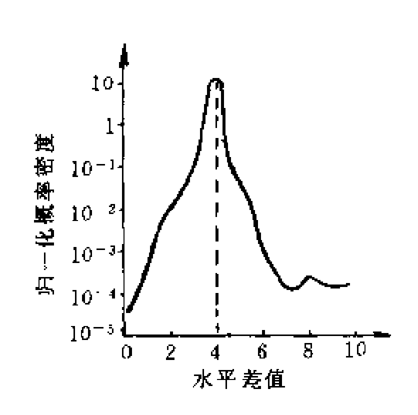
图 2-1　图像振幅分布密度曲线

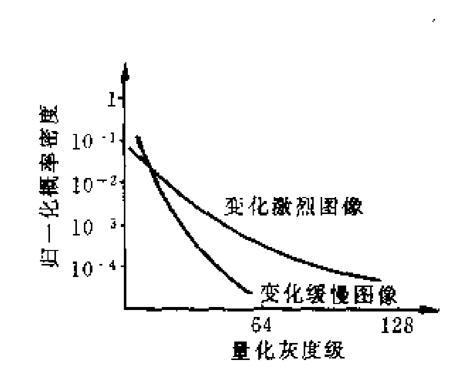
2. 差值信号的振幅分布特性

对图像信号预测编码有重要意义的差值信号的振幅分布特性也进行了测定。早在 1952 年，贝尔实验室的克雷茨默(Kretzmer)对相邻像素间的差值进行了测定，其分布密度曲线如图 2-2 所示，由曲线可见，相邻像素振幅的差大部分集中于零差值附近。这说明相邻像素间有较大的相关性。由曲线的形状，可认为其近似于指数分布，即

$$f(x)=e^{-\alpha x} \tag{2-11}$$

式中 x 是像素间的距离，α 是由图像性质决定的系数，$f(x)$ 是概率密度函数。由实测结果可知，对于特写画面来说 α 值较小，对于群集远景画面来说 α 较大。





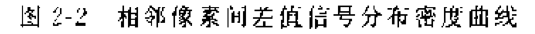


图 2-2　相邻像素间差值信号分布密度曲线

图 2-3　帧差值信号分布密度特性

对于电视信号来说，除了像素间的差值外，还存在帧间差值。Candy 对帧差值信号的分布密度特性也进行了测定，其曲线如图 2-3 所示。由图 2-3 可见，变化剧烈的图像与变化缓慢的图像其差值的分布是不一样的。帧差值信号分布密度大致也可以用指数函数来表示，即

$$f(x)=e^{-\beta x} \tag{2-12}$$

式中 β 是由图像变化程度所决定的系数。当图像内容变化比较剧烈时，β 值较小；图像内容变化比较缓慢时，β 值较大。

3. 图像的自相关函数和空域功率谱

一般情况下，图像的自相关函数和空域功率谱针对稳定画面来定义。稳定画面就是指图像 $g(x,y)$，在 x,y 方向上可无限扩展，而且无论在哪个方向上都有相同的统计特性。为了充分了解这样的稳定画面的统计特性，我们将研究有限的、固定的图像之统计特性。

(1) 能量有限的图像振幅谱和自相关函数

当一幅图像用 $g(x,y)$ 来表示的话，那么它的傅里叶变换就是其振幅谱，即

$$G(u,v)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}g(x,y)\mathrm{e}^{-j(ux+vy)}\mathrm{d}x\mathrm{d}y \tag{2-13}$$

显然，如果知道了图像的振幅谱，用其反傅里叶变换就可以求得 $g(x,y)$，

$$g(x,y)=\frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}G(u,v)\mathrm{e}^{j(ux+vy)}\mathrm{d}u\mathrm{d}v \tag{2-14}$$

图像信号所具有的能量可由下式表示：

$$E=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|g(x,y)|^2\mathrm{d}x\mathrm{d}y \tag{2-15}$$

或者是

$$E=\frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|G(u,v)|^2\mathrm{d}u\mathrm{d}v \tag{2-16}$$

式中 $|G(u,v)|^2$ 为能量谱。对于能量有限的图像信号 $g(x,y)$ 的自相关函数由下式表示：

$$\rho(\xi,\eta)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}g(x+\xi,y+\eta)g(x,y)\mathrm{d}x\mathrm{d}y \tag{2-17}$$

其中 $\rho(\xi,\eta)$ 与 $|G(u,v)|^2$ 之间存在着傅里叶变换关系，即

$$|G(u,v)|^2=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\rho(\xi,\eta)\mathrm{e}^{-j(u\xi+v\eta)}\mathrm{d}\xi\mathrm{d}\eta \tag{2-18}$$

$$\rho(\xi,\eta)=\frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|G(u,v)|^2\mathrm{e}^{j(u\xi+v\eta)}\mathrm{d}u\mathrm{d}v \tag{2-19}$$

(2) 功率有限的图像的空域功率谱和自相关函数

作为振幅谱和能量谱概念的自然扩展就可以导出功率谱的概念来。振幅谱或能量谱是对能量有限的画面定义的。但是，多数画面在 (x,y) 平面上是无限扩展的。因此，其能量 E 也不可能是有限的。对于 E 为无限的信号来说，有时图像中单位面积上的平均能量（平均功率）却是有限的，即：

$$W=\lim_{X,Y\to\infty}\frac{1}{XY}\int_{-\frac{X}{2}}^{\frac{X}{2}}\int_{-\frac{Y}{2}}^{\frac{Y}{2}}|g(x,y)|^2\mathrm{d}x\mathrm{d}y \tag{2-20}$$

如果把 $g(x,y)$ 限定在如下范围内的话，即

$$g_{XY}(x,y)=\begin{cases}g(x,y) & -\dfrac{X}{2}\leqslant x\leqslant\dfrac{X}{2},\ -\dfrac{Y}{2}\leqslant y\leqslant\dfrac{Y}{2}\\ 0 & \text{其他}\end{cases}$$

则式(2-20)可写成下面的形式：

$$W=\lim_{X,Y\to\infty}\frac{1}{XY}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|g_{XY}(x,y)|^2\mathrm{d}x\mathrm{d}y \tag{2-21}$$

如果令 $g_{XY}(x,y)$ 为有限能量的图像，其对应的振幅谱就是 $G_{XY}(u,v)$。此时则有

$$XYW \approx \frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|G_{XY}(u,v)|^2\mathrm{d}u\mathrm{d}v \tag{2-22}$$

$$W \approx \frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{XY}|G_{XY}(u,v)|^2\mathrm{d}u\mathrm{d}v \tag{2-23}$$

当 X,Y 无限增大时，功率谱密度 $\psi(u,v)$ 如下式表示：

$$\psi(u,v) = \lim_{X,Y\to\infty}\frac{1}{XY}|G_{XY}(u,v)|^2 \tag{2-24}$$

此时式(2-23)可写成如下形式：

$$W \approx \frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\psi(u,v)\mathrm{d}u\mathrm{d}v \tag{2-25}$$

当能量无限增大时，可定义自相关函数如下：

$$\Psi(\xi,\eta) = \lim_{X,Y\to\infty}\int_{-\frac{Y}{2}}^{\frac{X}{2}}\int_{-\frac{X}{2}}^{\frac{Y}{2}}g(x+\xi,y+\eta)g(x,y)\mathrm{d}x\mathrm{d}y \tag{2-26}$$

这里，$\psi(u,v)$ 和 $\Psi(\xi,\eta)$ 的关系是一对傅里叶变换关系，即

$$\psi(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\Psi(\xi,\eta)\mathrm{e}^{-\mathrm{j}(u\xi+v\eta)}\mathrm{d}\xi\mathrm{d}\eta \tag{2-27}$$

$$\Psi(\xi,\eta) = \frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\psi(u,v)\mathrm{e}^{\mathrm{j}(u\xi+v\eta)}\mathrm{d}u\mathrm{d}v \tag{2-28}$$

(3) 自相关函数的基本性质

$$\Psi(0,0) = E\{|g(x,y)|^2\} \tag{2-29}$$

$$\Psi(\infty,\infty) = \{E\{g(x,y)\}\}^2 \tag{2-30}$$

$$\Psi(x,y) = \Psi(-x,-y) \tag{2-31}$$

(4) 空域功率谱密度的基本性质

$$\psi(0,0) = \frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\psi(u,v)\mathrm{d}u\mathrm{d}v \tag{2-32}$$

$$\psi(u_x,u_y) = \psi(-u_x,-v_y) \tag{2-33}$$

(5) 自相关函数的测定结果

图像的自相关函数最早也是由贝尔实验室的克雷茨默测定的。测定采用光学方法，用两张完全相同的幻灯片，测量两者稍微错开时的透光量和完全重叠时的透光量，然后求其比值，进而得到相关函数。测定的曲线如图 2-4 所示。由图可见，自相关函数也近似于指数分布，即

$$\Psi(\xi,\eta) = \exp(-\alpha\sqrt{\xi^2+\eta^2}) \tag{2-34}$$

式中，α 是与画面有关的系数，对于特写画面 α 较小，群集的远景较大。

这一性质日本的矶部、藤村也同样用光学方法进行了测试，所得结果基本相似。

另外，克雷茨默对一帧场景各个方向上的相关性的测定，发现并没有明显的相关方向，其测试结果如图 2-5 所示。

4. 电视信号的数学表示及自相关函数

在数字信号处理中取样定理是大学所熟悉的基本定理。对于一维信号来说，如果 $x(t)$ 是频带受限信号，它所包含的最高频率为 f_c，当取样频率满足 $f_s \geqslant 2f_c$ 时，则完全可由其离散样点值来表示，即

$$x(t) = \sum_{n=-\infty}^{+\infty}x(nT)\frac{\sin\omega_c(t-nT)}{\omega_c(t-nT)} \tag{2-35}$$

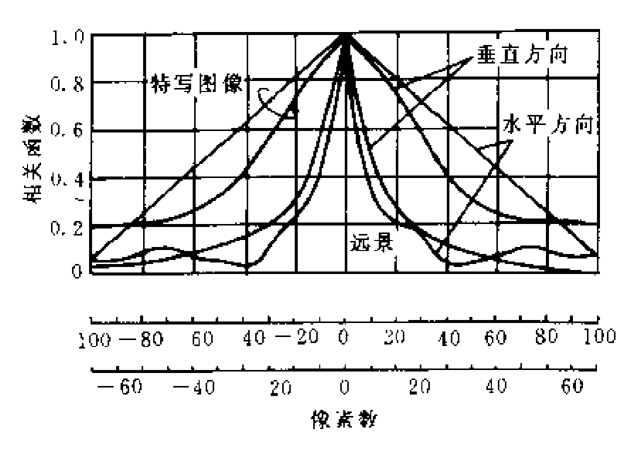


图 2-4 图像在水平和垂直方向上的自相关函数

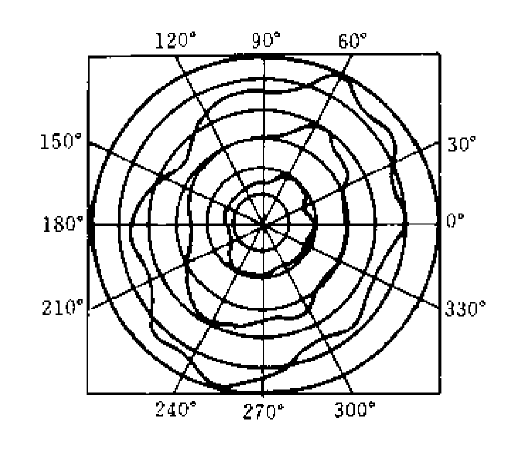


图 2-5 某景物各个方向上自相关函数的轮廓

式中，$\omega_c=2\pi f_c$、$T=\frac{1}{f_s}=\frac{1}{2f_c}$，$\frac{\sin\omega_c(t-nT)}{\omega_c(t-nT)}$为取样函数或称内插函数。

仿照一维取样定理的结果，可以将其推广到二维情况。如果$x(t_1,t_2)$是频带受限的二维信号，在t_1方向上所含最高频率为ω_1，在t_2方向上所含最高频率成份是ω_2，在t_1方向上的取样间隔为T_1，其中$T_1=\frac{\pi}{\omega_1}$，在t_2方向上的取样间隔为T_2，其中$T_2=\frac{\pi}{\omega_2}$。则

$$x(t_1,t_2)=\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}x(iT_1,jT_2)\cdot\frac{\sin\omega_1(t_1-iT_1)}{\omega_1(t_1-iT_1)}\cdot\frac{\sin\omega_2(t_2-jT_2)}{\omega_2(t_2-jT_2)} \tag{2-36}$$

如果是动态图像，则可以推广到三维情况，也就是：

$$x(t_1,t_2,t_3)=\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty}x(iT_1,jT_2,kT_3)\cdot\frac{\sin\omega_1(t_1-iT_1)}{\omega_1(t_1-iT_1)}\cdot\frac{\sin\omega_2(t_2-jT_2)}{\omega_2(t_2-jT_2)}\cdot\frac{\sin\omega_3(t_3-kT_3)}{\omega_3(t_3-kT_3)} \tag{2-37}$$

根据动态图像采样定理可以算出动态图像所需的带宽。例如，设有一帧图像的扫描时间为T_p，在t_1、t_2方向上像素间隔均为T，而且在t_1、t_2方向上各需N_1、N_2个像素，那么

$$T_p=N_1N_2T \tag{2-38}$$

由取样定理可知，$T=\frac{1}{2f_c}$，所以最高频率$f_c=\frac{1}{2T}$，因此，

$$f_c=\frac{1}{2T}=\frac{N_1N_2}{2T_p}=\frac{1}{2}N_1N_2f_p \tag{2-39}$$

一般情况下，在电视体制中要考虑到其他因素，所以最高频率由下式计算：

$$f_c=\frac{1}{2}KN_2^2f_pRR' \tag{2-40}$$

式中，K为凯尔(kell)系数，通常$K=0.7$；R为电视屏幕之宽高比，$R=\frac{4}{3}$；R'为水平扫描率与垂直扫描率之比，通常$R'=\frac{0.95}{0.84}$；N_2是扫描行数；f_p是帧频数。

例如，在NTSC制中，$N_2=525$，$f_p=30$，则可算出其最高频率

$$f_c=\frac{1}{2}\times0.7\times(525)^2\times30\times\frac{4}{3}\times\frac{0.95}{0.84}=4.3\text{MHz}$$

动态图像的自相关函数可用下式表示：

$$R(t_1,t_2,\tau)=\exp[-\alpha_1|t_1|]\cdot\exp[-\alpha_2|t_2|]\cdot\exp[-\beta|\tau|] \tag{2-41}$$

式中，$\exp[-\alpha_1|t_1|]$中表示行内相关；$\exp[-\alpha_2|t_2|]$表示帧内相关；$\exp[-\beta|\tau|]$表示帧间相关。其中β是反映时间轴方向上的相关性，研究表明，β是一个很小的量，一般$\exp[-\beta|\tau|]=0.99$，也就是说帧间的相关性接近于1。

图像自相关的另外一种形式是用一阶马尔可夫过程作为模型(在第5章中再详细介绍)。在图像自相关函数的两种模型中，应用较多的是根据大量实验总结出来的e指数模型，即式(2-34)和式(2-41)所示的形式。至于其衰减的速度完全取决于图像的种类及其内容细节。

2.1.5 图像信息的信息量

1. 离散的图像信息的熵

对于一个连续的图像信号经过编码后就变成了离散的图像信息。一幅图像如果有$s_1,s_2,s_3,\cdots,s_q$共q种幅度值，并且出现的概率分别为$P_1,P_2,P_3,\cdots,P_q$，那么每一种幅值所具有的信息量分别为$\log_2\left(\frac{1}{P_1}\right),\log_2\left(\frac{1}{P_2}\right),\log_2\left(\frac{1}{P_3}\right),\cdots,\log_2\left(\frac{1}{P_q}\right)$。由此，其平均信息量可由下式表示：

$$H=\sum_{i=1}^{q}P_i\log_2\left(\frac{1}{P_i}\right)=-\sum_{i=1}^{q}P_i\log_2 P_i \tag{2-42}$$

把这个平均信息量叫做熵，记作H。

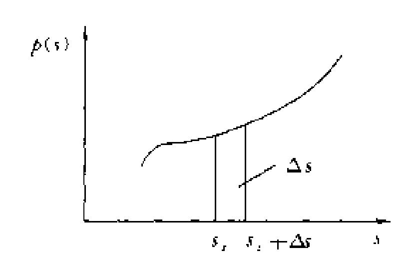


图2-6 连续信源的熵的计算

如果一个图像信源能输出K个独立的消息，当这些消息出现的概率彼此相等时，那么这个信源的熵最大。例如，一个信源只输出P和$P-1$两个消息，熵的最大值出现在两个消息的概率都等于0.5处。

2. 连续的图像信息的熵

对于离散的图像信息来说，它只输出有限个符号。如果输出的不是有限个而是无限个符号，那么，这样的图像信息叫做连续图像信息。对于连续图像信息的熵也可以仿照离散图像信息的熵来计算。如图2-6所示的连续信源，把s分成小微分段Δs，这样，类似于离散信源的熵可导出如下：

$$H=-\sum_{i=-\infty}^{+\infty}p(s_i)\cdot\Delta s\cdot\log_2[p(s_i)\Delta s]=\sum_{i=-\infty}^{+\infty}p(s_i)\cdot\Delta s\cdot\log_2\frac{1}{p(s_i)\Delta s}$$

$$=\sum_{i=-\infty}^{+\infty}p(s_i)\cdot\Delta s\cdot\log_2\frac{1}{p(s_i)}+\sum_{i=-\infty}^{+\infty}p(s_i)\cdot\Delta s\cdot\log_2\frac{1}{\Delta s}$$

当$\Delta s\to 0$时，则

$$H\approx\int_{-\infty}^{+\infty}p(s)\log_2\frac{1}{p(s)}\mathrm{d}s+\infty$$

第二项是由于$\Delta s\to 0$时，$\log_2\frac{1}{\Delta s}\to\infty$所致。一般忽略掉第二项，连续的图像信息源的熵如下式：

$$H=-\int_{-\infty}^{+\infty}p(s)\log_2 p(s)\mathrm{d}s \tag{2-43}$$

这里应该注意的是连续图像信息的熵并不是绝对熵，而是绝对熵减去一个无限大项。因此，也可以说这是一个相对熵。其中$p(s)$是概率密度。

对于离散信源来说，当所有消息输出是等概率时其熵最大。但对连续信源来说最大熵的条件取决于输出受限情况。当输出幅值受限的情况下，幅度概率密度是均匀分布时其熵值最大。

当输出功率受限的情况下，则输出幅度概率密度是高斯分布时其熵值最大。关于熵的概念在图像编码处理中有重要意义。

2.2 图像处理系统及外围设备

数字图像处理技术的进展及其日益广泛的应用促进了处理系统硬件设备的研制与开发。在图像处理技术发展的历史中，最初在模拟方案识别机构中开始应用计算机。它处理的对象是二值图像，处理的像素数目较少，数据量不大。因此，处理速度问题并没有显得十分突出。尽管如此，也看到了图像处理与一般的科学计算有着显著的不同，就是在图像处理中应该保持其二维的特点。图像的数目越来越多，每幅图像的像素数目越来越大，因此，处理一幅图像所需的时间越来越长，加之大型机的字长较长、代价较高、效率较低，这就愈来愈明显地看出通用计算机的图像处理能力有一定限制。因此，人们开始研制专用的计算机图像处理系统。到目前为止，已出现了各种各样的计算机图像处理系统，尽管各种系统大小不一，其处理能力也各有所长，但其基本硬件结构则都是由如图 2-7 所示的几个部分组成的，即由主机、输入设备、输出设备及存储器组成的。目前，这些系统就其应用领域来看有专用的，也有通用的。它们的主要差别则在于处理精度、处理速度、专用软件及存储容量等几个方面。为了较系统地理解图像处理系统的硬件设备，本节主要介绍几种常见的输入、输出设备的工作原理及图像处理系统的典型结构。

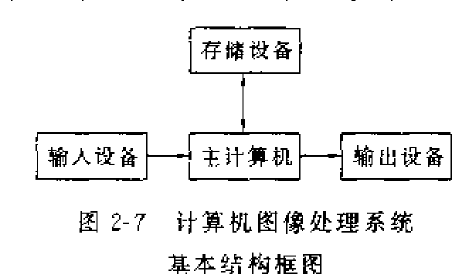


图 2-7 计算机图像处理系统基本结构框图

2.2.1 图像处理系统中常用的输入设备

图像处理的输入设备有很多种，其工作原理及性能也各有长短。常用的输入设备主要有如下几种：

1. 电视摄像机

电视摄像机是图像处理中常用的输入设备之一，它的关键部件是摄像器件。摄像器件的基本任务是把输入的二维辐射(即光学图像)信息转换为适宜处理和传输的电信号。

(1) 摄像器件的分类

目前，由于对摄像器件不同用途的需要，发展的品种极为繁多，大体上可作如下分类：

① 按输入信息分类　可分为可见光方式(黑白或彩色方式)；不可见电磁辐射方式(红外线、紫外线及 X 射线)及超声波等。

② 按电流方式分类　可分为电子束扫描摄像管(快电子束、慢电子束和光束扫描)；固体摄像器件(XY 寻址扫描，电荷传输扫描及弹性表面波扫描)。

③ 按光敏靶的种类分类　可分为光电导型(注入型、阻挡型)；移像型(低增益靶，如超正析像管、分流直像管及二次电子传导等，电子轰击导生电导)；特殊型(热释电及压电型)。

(2) 摄像器件的性能

摄像器件的性能好坏可以从以下几个方面来考虑：

① 频谱响应特性　亦称光谱响应特性、分光灵敏特性、光谱灵敏度等。它表示器件的分光灵敏度与相应波长的关系。

② 辐射灵敏度　也称响应度，即单位辐射通量或单位辐射照度均匀地输入到器件的接收

靶面时所能产生的输出信号电流的大小。

③ 光电变换特性　亦称光信号变换特性。它是反映输入辐射或光量发生变化时相应的输出信号电流的变化。其中一个重要参数就是 r 值，一般常选用 r 值小于 1 的摄像器件。

④ 信噪比　摄像器件噪声的来源很多，不同的探测目的所要求的输出信噪比也不一样。一般在广播电视中要求信噪比在 40dB 以上。在普通摄像器件中，内部噪声是主要的，它主要来源于电流的散弹噪声。

⑤ 暗电流　指在没有输入信息时器件输出的电流。从理论上讲，如果暗电流在空间上和时间上是恒定的，它并不构成噪声。但是，实际上它不仅增加了电流噪声，而且会使电子束发射系统和前置放大器负担过重。如果暗电流在空间和时间上不是恒定的话，那就会直接构成噪声，增加了图像亮度的不均匀性和色平衡困难。

⑥ 分辨率　指器件对图像细节的鉴别能力。通常是把一高对比度的鉴别率测试图案投射到光接收面上，然后观察可分辨的最小空间频率数。一般用可分辨电视行数(TVL/H)或(lP/mm)为单位来度量。水平条纹反映的是垂直分辨率，它受扫描线数限制。垂直条纹反映的是水平分辨率。在有足够的频带宽度的条件下，基本决定了摄像器件的分辨能力。因此，测试器件时主要看垂直条纹，应注意到 CPT(彩色显像管)和 CDT(彩色显示管)的差别。

⑦ 调制传递函数和方波振幅响应度　在工程上更严格的表征器件分辨能力的方法。

⑧ 惰性　指输入信息在强度发生变化时，输出信号的相应变化在时间上的滞后现象。它有起始惰性和衰减惰性之分，还可分为电容性惰性和光电导惰性。它们分别与阴极温度高低、靶材料物理特性和器件的工作条件等有关。

⑨ 动态范围　摄像器件动态范围有两种含义：一是指器件能够处理的一帧景物内最亮单元和最暗单元的亮度的比值。它反映器件能探测的最小信息和能接受的最大信息的能力；另一方面指的是器件能容纳的不同的平均景物亮度的范围，即最亮的一帧景物平均亮度和最暗一帧景物平均亮度的比值。有时把前者称为内动态范围，后者称为外动态范围。

除了上面所列的评价摄像器件的性能的几项指标外，还有如畸变、疵点、晕光、成荫、余像烧伤、寿命、机械强度和环境适应性等等。

以上对摄像器件的基本分类和性能作了简要的介绍，下面介绍几种常用的摄像器件的结构与特性，以便在建立图像处理系统时，根据不同的应用目的加以合理地选用。

(3) 几种常用的摄像器件

① 光电导摄像管

光电导摄像管亦称视像管。其中硅靶管、碲化锌镉管和硒化镉管属于微光摄像器件，而硫化锑管、氧化铅管及硒砷碲管灵敏度较低。视像管的结构如图 2-8 所示。

由图 2-8 可见，视像管主要由玻璃壳、光导靶及电子枪组成。电子枪及其附属的聚焦线圈、偏转线圈及校正线圈一起组成了电子光学系统。它的主要任务是形成电子束并对之进行聚焦和偏转。对电子束的要求是：电子束直径小、密度大、扫描线性好、垂直上靶。图中的热丝和阴极是发射电子的源，它可以是直热式的也可以是傍热式的。调制栅极的任务是会聚阴极发射的电子并控制束流的大小。加速极加有＋300V 左右的电压，它使阴极发射的电子得到加速，并且与调制极一起形成电子透镜，从而使电子束受到会聚。聚焦部分的作用是使到达靶面的电子束聚成一锐点。它是由聚焦电极和加速电极构成的电子透镜完成这个任务的。在聚焦电极外还有一长聚焦线圈，它构成一长磁透镜。采用磁聚焦可以使图像中心分辨率提高，但体积、重量、功耗都较大。为此也可以采用纯静电聚焦结构。静电聚焦的缺点是中心分辨率有所降低，

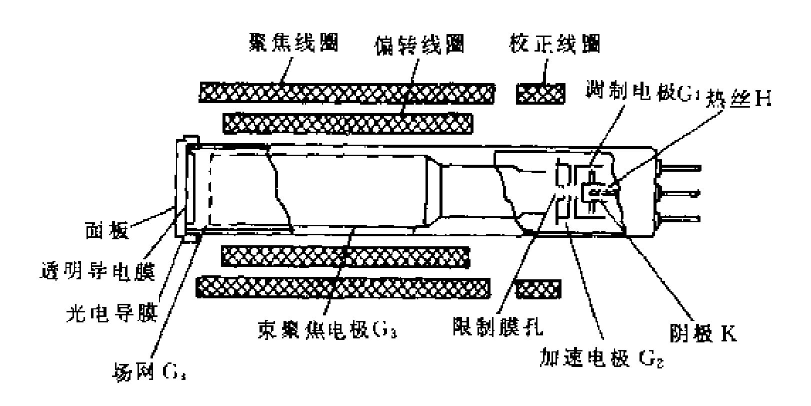


图2-8 视像管的结构示意图

但由于畸变小、信号输出均匀性较高、体积小、重量轻、功耗低，因此经常使用。

偏转部分的作用是使电子束按一定规律偏转以形成扫描光栅。它既可用磁偏转也可以用静电偏转。虽然静电偏转方式的设备体积、重量及功耗都比较小，但磁偏转的磁场均匀、像差小、偏转灵敏度高、偏转线性好，因此，目前绝大多数摄像管还是采用磁偏转方式。

准直部分的作用是使聚焦后的电子束垂直上靶。它是通过在靶附近设置一个场网及在调制极附近加校正线圈来完成的。由于场网的作用，使得靶和网之间产生强烈的垂直减速场，减少了横向电场的影响，从而改善了正交性。

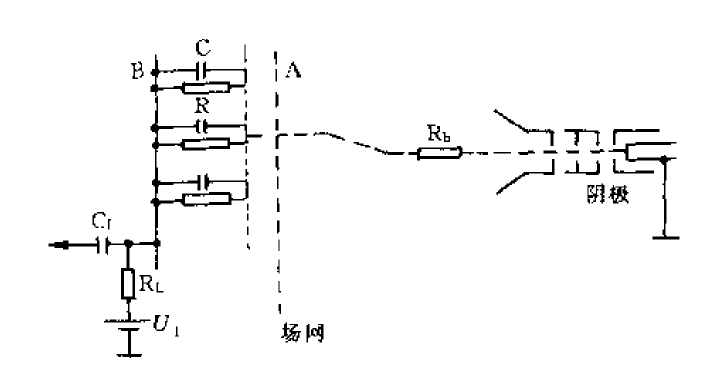


图2-9 视像管产生视频信号的等效电路

视频信号的产生过程可由图2-9的等效电路来说明。设靶共包含有n个像素，每个像素可简单地等效于一个电阻R和一个电容C的并联。从阴极发射出来的电子束通过场网后进入减速场，以慢速落在靶的扫描侧A上。由于靶压U_T很低，二次电子发射系数$\sigma<1$，因此，进入靶的电子比出来的电子多，最后，使扫描侧A的电位等于阴极电位。这时电源U_T，负载电阻R_L，电容C和阴极构成通路，电容充电。设在无光照时某一像素的暗电阻为R_d，当电子束离开像素后，电容器开始沿R_dC回路放电，但不输出信号电流。靶的外侧B电位固定为靶压U_T，扫描侧A的电位U_{Ad}随像素电容器C的放电而上升，即

$$U_{Ad}=U_1\left[1-e^{-\frac{t}{R_dC}}\right] \tag{2-44}$$

像素电容的放电时间近似于帧周期T_f，因此，在下一次对像素进行扫描充电之前，扫描侧A点的电位最大值为

$$U_{Adm}=U_1\left[1-e^{-\frac{T_f}{R_dC}}\right] \tag{2-45}$$

因为暗电阻R_d非常大，所以，$U_{Ad}\approx 0$。在两次扫描间隔期内，电容A侧电位上升量很小。

在下一次扫描时，A 侧电位又恢复为阴极电位。这时的充电电流就称为暗电流。当靶面各像素不一致时，暗电流也不一样。当充电电流经过束电阻 R_b，像素电容 C，负载电阻 R_L，靶电源 U_T 到地时，在 R_L 上产生电压降 ΔU_d，这个电压变化经耦合电容 C_t 耦合到视频前置放大器，就是黑色电平的视频信号。

在有光照时，电阻率减小，像素电阻 R 变为光电阻 R_c。照度越大，则 R_c 越小，这时 R_cC 回路的放电速度越快，在两次扫描间隔期内，靶 A 侧的电位上升量越大。反过来，照度越小，像素电阻 R 越大，靶 A 侧电位上升量越小。由此，充电时在负载电阻 R_L 上就形成一个与像素照度相对应的电压降，这个电压降耦合到前置放大器，经放大后便可输出一个视频信号。

视像管的电子枪部分都是一样的，不同的靶就构成不同种类的视像管。靶是由具有光电效应的半导体材料制成，它的任务就是把输入的光学图像变成为电荷图像。光电导靶的材料应满足如下一些条件：能响应在工作波段的入射光子，灵敏度高，暗电流小，响应快，衰减惰性小，能保证必要的分辨能力等。下面择要介绍几种用不同的半导体材料制成的视像管及其性能，以便在使用中根据不同的需要加以合理的选择。

(a) 硫化锑视像管

硫化锑视像管是注入型靶的代表。由于是注入型靶，所以其缺点是灵敏度低、惰性大、红光响应差、暗电流大、受温度影响较大。而且其 r 值远小于 1，也不宜用做彩色摄像机。它的主要优点是靶的成份和结构较简单，性能稳定。因此比较容易制造，成品率高。此外，由于 r 值远小于 1，因此有较宽的动态范围。如 8507 型 1 英寸硫化锑视像管的灵敏度为 200nA/10lx；极限分辨率为 1000TVL；暗电流为 20nA；r 值为 0.65；工作温度为 +10～+40℃。

(b) 氧化铅视像管

氧化铅视像管是阻挡型结构的代表。它的光灵敏度比硫化锑管高一些，但辐射灵敏度与波长有关。在靶压 U_T 大于 30V 以后，光电流基本不受靶压的影响。这种视像管暗电流极小，惰性也很小。由于靶较厚，所以其分辨率略低于硫化锑管。由于有上述一些优点，氧化铅视像管适合应用于彩色摄像机中，但由于氧化铅在空气中不稳定、成品率较低，因此，成本高、价格昂贵，从而限制了它的应用范围。XQ1080 型氧化铅视像管的灵敏度为 375μA/1m；分辨率为 750TVL，暗电流为 0.5～1.5nA，r 值为 0.95，工作温度为 -30～-50℃。

(c) 硅靶视像管

用几十万个硅二极管阵列靶代替硫化锑靶，即构成硅靶视像管。制造硅靶的工艺和大规模集成电路的工艺一样，它是在 N 型硅片的一面经抛光、氧化形成一层绝缘良好的二氧化硅(SiO_2)薄膜，然后经光刻、硼扩、减薄、磷扩、退火和腐蚀等工艺过程制成称为电阻海的半绝缘层。其结构如图 2-10 所示。

在二氧化硅薄膜上刻有许许多多窗孔，经过扩硼工艺形成一个个 P 型岛。每个 P 型岛与 N 型基片构成一个 PN 结二极管，它被二氧化硅薄膜隔开，这样，就在 N 型硅片上形成一个 P 型岛阵列。当电子束扫描时，由于有电阻海的存在，可以使电荷流向各个二极管。在 N 型基片上加上比阴极电位高的电位，使二极管反向偏置。当无光照射时，暗电流很小。在有光照射时，在 N 型区中就产生电子空穴对，空穴向电位低的 P 区移动，使靶扫描侧的电位升高，其升高的数值与光照强度成正比。当靶受到电子束扫描时，其电位被拉向阴极电位，这相当于充电过程，从而在负载中形成与光学图像相对应的图像信号。

硅靶管的优点是灵敏度较高、惰性小、光电特性较理想、r 系数近似于 1，而且硅靶不易被烧伤、使用寿命较长。

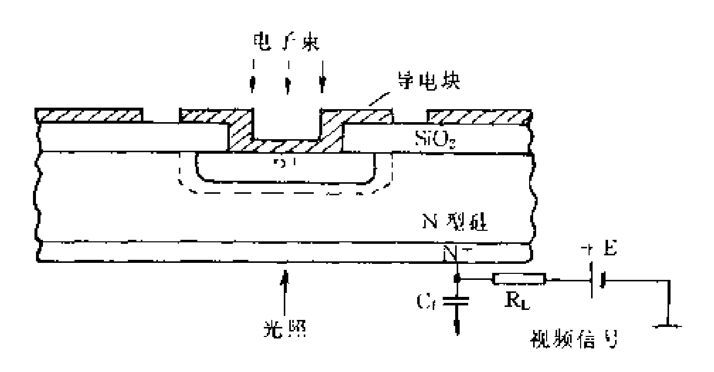


图 2-10 硅靶结构与工作原理

硅靶管工作靶压的选择对摄像的性能影响较大。当工作靶压较高时，耗尽层加宽，电容变小，因而惰性变小，并且电子束上靶容易，信号电流增大，边缘像质有所改善。但此时暗电流加大，疵点增多和变大。当工作靶压较低时，暗电流减小，疵点变少变小，但惰性变大，输出电流变小。在靶压过低时还会造成暗电流不均匀，以致产生不均匀的背景。通常工作靶压确定在 8～12V 左右。在 30℃时，暗电流约为 8～12nA，温度每上升 9℃，暗电流约增加一倍。RCA4532 管光谱响应较宽，可达到 1.1μm，因此，可用作近红外摄像管。它的灵敏度为 4350μm/1m 或 250nA/0.5lx、中心极限分辨率为 700TVL/H，暗电流为 10nA，r 值近似等于 1，抗烧伤能力极好，工作温度为－40～＋50℃。

(d) 硒化镉视像管

硒化镉视像管的靶是所谓异质复合结靶。硒化镉视像管的光电灵敏度很高，普通型靶达 2670μA/1m，而且对长波，紫外线也有很好的响应，可直接制成紫外摄像管。它的另一个特点是暗电流极小，可小于 1nA。特别是采用 ZnS 作电阻膜时，暗电流更小，可在数十秒之内积累信号电荷而基本上不降低清晰度，因此可采用慢扫描或断续扫描方式，这在图像处理系统中是有用的。E5063 型（东芝产品）视像管灵敏度为 2600μA/1m 或 160nA/0.5lx，暗电流约 1nA，极限分辨率为 800TVL，r 值为 0.95，抗烧伤能力很好，工作温度为－20～＋60℃。

(e) 硒砷碲视像管

硒砷碲视像管的特性在各方面都和氧化铅管十分相似，而它的工艺性能稳定，成品率高，所以价格比氧化铅管便宜得多，故而应用较广。日立公司的 9326 型视像管灵敏度如下：W 为 350μA/1m，R 为 120μA/1m，G 为 150μA/1m，B 为 80μA/1m。极限分辨率为 1100TVL，r 值近似等于 1，暗电流为 0.6nA，工作温度＋25～＋35℃。

(f) 碲化锌视像管

碲化锌视像管的光电灵敏度和硅靶视像管差不多，但光谱灵敏度有些差别，在可见光谱范围内，比硅靶管高 1.5～2 倍。此外，晕光现象比硅靶小，工艺也比较简单，价格比较便宜。但其红外灵敏度、抗烧伤和惰性等方面比硅靶管差。碲化锌视像管适用于低照度监视及其他弱光电视中。松下 S4076 型管灵敏度为 240nA/0.5lx，极限分辨率为 800TVL，暗电流为 8nA(25℃)，r 值为 1，抗烧伤能力极好，工作温度≤70℃。

综上所述，各种光电导摄像管的性能列于表 2-2 中。以便对它们的性能作一比对。

表 2-2　各种视像管典型性能一览表

性能＼管名	硫化锑管	氧化铅管	硅靶管	硒化镉管	硒砷碲管	碲化锌镉管
光电灵敏度(μA/lm)	170	350～400	4350	2600	350	4300
极限分辨率(TVL)	800	750	700	800	900	800
惰性(三场后)(%)	20～25	～3	7～10	10～20	<2	<20
暗电流(nA)	20	2	10	<1	<1	10
γ值	0.65	0.95	1	0.95	1	1
动态范围	350∶1	60∶1	50∶1	60∶1		
抗烧伤能力	差	较差	极好	很好	较好	较好
晕光现象	很小		严重	小		很小
工作温度(℃)	+10～+40	−30～+50	−40～+50	−20～+60	+25～+35	<70

② 移像型摄像管

超正析像管、分流直像管、二次电子传导管及电子轰击导生电导管均属移像型摄像管。它们大都属于微光摄像器件。

(a) 超正析像管(IO)

超正析像管又叫移像直像管，其结构如图 2-11 所示。

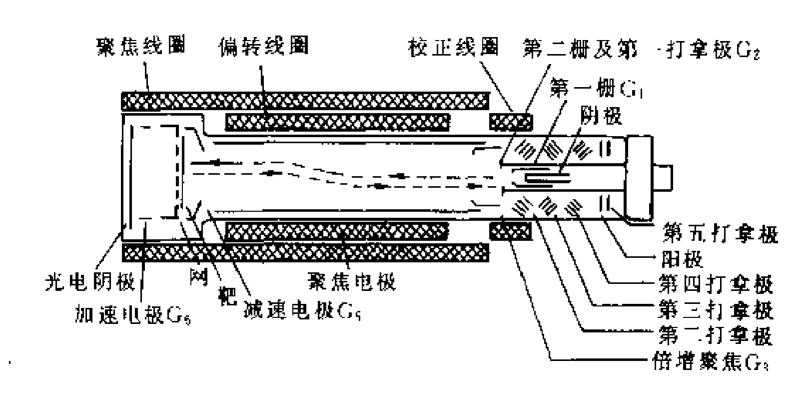


图 2-11　超正析像管结构示意图

超正析像管的结构可分为三个部分，即移像段、扫描段和电子倍增器。光学图像投射到光电阴极上，产生相应强度的光电子，在加速极电场和聚焦线圈磁场的共同作用下打到由钠玻璃制成的靶上，产生二次电子，在靶上形成了电位起伏，从而使光电阴极上的电位像平移到了靶上。二次发射系数越大，移像后的电位起伏也扩大得越大。加速极电压的选择与聚焦磁场的大小要保证光电子的会聚点正确地落在靶上，并使σ尽可能地大。靶内产生的二次电子被前面相距 50μm 处带有正电位的网所收集，因而在靶上出现与输入光电子相应的正电荷图像。同时，钠离子携带正离子扩散到靶的扫描侧。扫描电子被该侧接受后，需要从这一侧传导到另一侧，以便将正电荷中和掉。因此，靶还必须有一定的直流导电性。由于靶较薄，横向电阻率高，它限制了电荷的横向扩散，因此分辨率较高。

电子枪发射的电子束穿过膜孔，在电场和磁场的共同作用下以螺旋线方式飞向靶。由于减速极和场网的作用，这些电子慢速上靶。

束电子被靶的起伏正电位所吸收，剩余电子被反射和散射，它们在减速场的作用下加速回

到加速阳极。靶的正电位越大，折回来的电子数越少，因此，折回的电子数反映了光像的强弱。但被反射和散射的电子已被散焦，因此，主要落在加速阳极旁边的电子倍增器的打拿极上，产生倍增二次电子。在圆筒电极的作用下这些倍增电子又落到第二打拿极上再次倍增，如此下去，经过四五次倍增，最后得到几百到几千的增益。

超正析像管的信噪比一般在 34dB 左右。超正析像管输出的视频信号电流大小与输入光的大小正好相反，输入光愈强，光电子愈多，靶电位上升高，吸收扫描电子多，返回的电子流就小，因而散弹噪声也相对地变小；而弱光时，回束电流及散弹噪声则相应地变大。这意味着在微光探测时，其信噪比劣化，它最多只能在 10^{-2}lx 的环境照度下工作。超正析像管的缺点除了信噪比较低外，尚有体积大、重量重、调节和维护麻烦等缺点。

(b) 分流直像管(IS)

分流直像管的移像部分和超正析像管一样，只是在扫描部分对返回的电子进行了不同的处理。当扫描电子束 I_p 上靶时，一部分束流 i_c 使靶充电后降到阴极电位；另一部分 i_s 落到靶上后反射回来；第三部分 i_r 未到靶就反射回来，其电子分化如图 2-12 所示，即

$$I_p = i_c + i_s + i_r \tag{2-46}$$

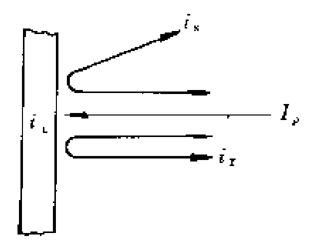




图 2-12　分流直像管内电子分化

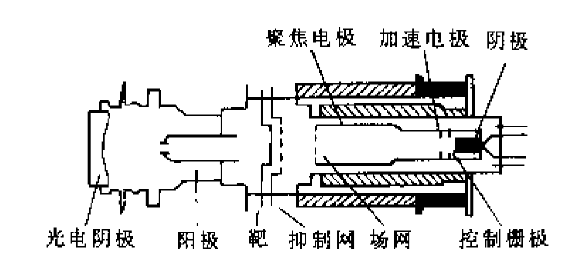


图 2-13　SEC 结构示意图

实验发现 i_s 与 i_c 成正比。与超正析像管不同，分流直像管是把 i_s 和 i_c 分开并仅仅使 i_s 得到放大。这样，不仅输出视频信号电流和输入光大小成正比，而且仅是分流部分的散弹噪声影响信噪比，故而可以提高微光下的输出信噪比。分流的方法可以采用加导向电极及分离电极的方法实现。

分流直像管在弱光时的信噪比可达 46dB，内动态范围很大，分辨率可达 1200TVL。它的惰性较小，光电灵敏度为 $5\times10^{2}\mu$A/lm。它的主要缺点是当面板照度下降时分辨率会迅速下降，此外，体积甚大，使用调节麻烦且苛刻，因此，它主要用于科研、医疗等方面。

(c) 二次电子传导摄像管(SEC)

二次电子传导摄像管是利用光电子产生二次电子传导作用进行电子增强。其结构示于图 2-13。SEC 管由成像段、靶和扫描段三个部分组成。光电阴极蒸发在光纤板的内壁。光电阴极按照投射在它上面的光学图像的亮度分布产生相应的光电子，在聚焦电场的作用下，高速打在二次电子传导靶上。在慢速扫描电子束的作用下，靶的扫描侧经常稳定在零电位，因此，靶的两端承受一定的靶电压。当一个从光电阴极飞出来的被加速的光电子打到靶上时，将在靶内产生二次电子。这些二次电子在靶电场的作用下流向信号板，在靶上留下一个正电荷图像，被扫描时，受电子束补充回到零位。此时，在外电路上产生的脉冲电流就是视频信息电流。从上述可见，它不是利用导带中的电子或满带中的空穴载流，而是由二次电子在电场的作用下传导，因此叫二次电子传导。这种管的灵敏度为 16000μA/lm，可在满月下工作；弱光时 γ 值接近于 1，照度较高时接近于 0.6；惰性较小；中心分辨率一般为 600TVL，具有较强的信息积累能力和

存储能力，暗电流小于 0.1nA，存储时间可达 24h。因此，在需要摄取相对静止的弱照度对象时，可以通过积累达到较高的信噪比。

(d) 电子轰击导生电导摄像管(EBIC)

利用高速电子轰击半导体或绝缘体，激发满带的大量电子到导带中去，形成许多自由电子-空穴对来获得高的增益，就叫做电子导生电导，简称 EBIC。其中最成功的是电子轰击电压硅靶摄像管，它的结构示于图 2-14。

从光电阴极发出的光电子在高电压的作用下轰击硅靶，在靶内激发电子空穴对，它与二次电子传导不同，是通过载流子在能带内运动载流的。这种摄像管的灵敏度约为 $3\times10^5\mu A/lm$；极限分辨率不超过 700TVL，其惰性、暗电流与硅靶摄像管相似。此外，管子寿命短一些。

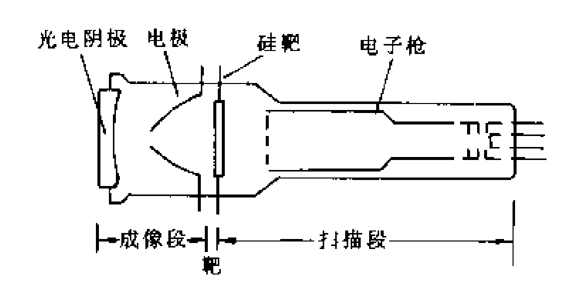


图 2-14 电子轰击硅靶摄像管结构示意图

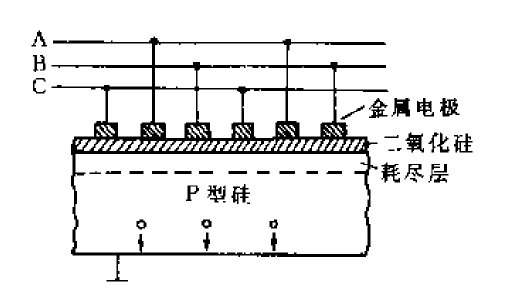


图 2-15 CCD 器件的基本结构

③ 固体摄像器件

固体摄像器件的研究始于 1960 年。1970 年，美国贝尔实验室发明了一种新型半导体器件-电荷耦合器件(CCD)。CCD 器件是一种 MOS 集成电路，基本结构如图 2-15 所示。

在 P 型硅衬底上通过氧化在表面形成一个二氧化硅层，然后在二氧化硅上蒸发一层金属膜，并用光刻的方法制成栅电极条。如果是三相驱动，那么，首先在所有金属栅极上都加上正偏压 U_1，U_1 要大于相应的绝缘栅场效应管的导通电压。这样就使电极下面的空穴远离二氧化硅表面，从而形成表面耗尽层，通常称为表面势阱。势阱的深度与所加的电压近似线性关系。如果在某一相加入比 U_1 更高的正电压 U_2，则在该金属电极下面就会形成更深的势阱，这种情况如图 2-16 所示。

这里，如果有光子入射就会产生空穴-电子对，少数载流子——电子就会存储在较深的势阱中，形成所谓电荷包。电荷存储情况如图 2-16(a)所示。由于栅极间的距离很小，因此，各栅极下的表面电势会发生耦合。如果在下一组栅极上加上更高的电压，那么，耗尽层的变化将如图 2-16(b)所示，电荷包会转移到更深的势阱中去。如果所加电压如图 2-16(c)所示，那么，电荷包就会转移到 B 相的栅极下面。由此原理，如果我们对 A、B、C 三相加以按时序变化的时钟脉冲电压，则电极下面的势阱也会按时序变化，电荷包也会从一端传输到另一端。最后通过一个反偏压输出二极管收集电荷并送入前置放大器。从上述原理可见，CCD 器件就是一个移位寄存器。

固体摄像器件从像素排列的结构上来看可分为二种类型：一是线阵摄像器件，一是面阵摄像器件。面阵摄像器件的帧传输方式如图 2-17 所示。

透镜将景物聚焦成像在光生电荷区，在场的正程，光生信号电荷进行积分，在场回扫时间内，光生电荷积分区都加帧转移高速时钟脉冲，使光生电荷区的整场图像快速地转移到存储区。在下一个场扫描正程时间内，存储区在行转移垂直寄存器的时钟脉冲驱动下，以一步一行

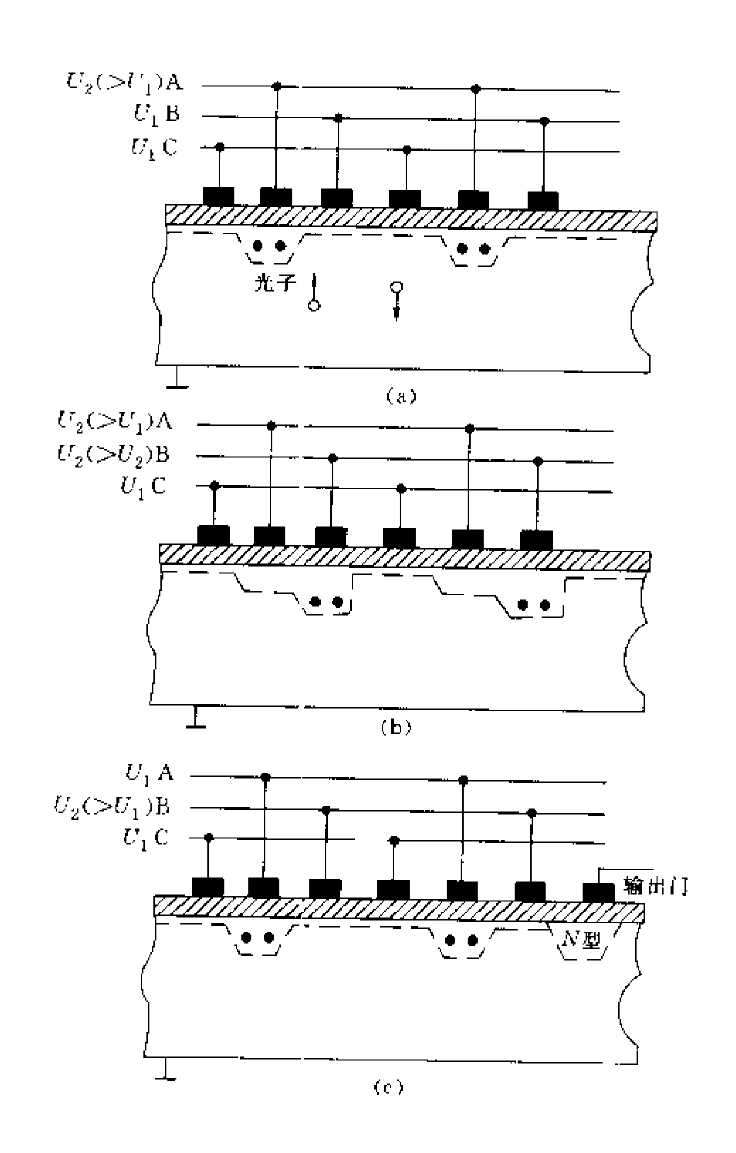


图 2-16　CCD 器件的基本结构

的方式，将整场光生信号电荷转移到水平寄存器内，每一行在一个行扫逆程内完成，各行之间的间隔为一个行扫正程。因此，正好在一场扫描正程时间内将存储区内全部信号电荷输出完毕，而这个时间也正好是新的图像信号电荷的积累时间。当下一场扫描逆程时间内将新的图像信号电荷转移到存储区时，存储区正好空着。对于输出水平寄存器来说，它在一个行扫正程时间内，在水平时钟脉冲的驱动下，将信号电荷逐位移到输出二极管，在行逆程时间内又接收下一行的信号电荷，然后陆续输出。就这样周而复始地输出视频信号。

CCD 的行间传输方式如图 2-18 所示。其结构是将传感元件和存储元件相间排列在光传感器件半透明多晶硅栅上，加上场积分脉冲以形成势阱，用以收集光生载流子。传感器件的列间垂直移位寄存器采用不透明的金属栅列，与传感器件一一对应。在一场积分完毕后，在垂直时钟脉冲电压的作用下传感器件内的信号电荷便转移到相邻的垂直寄存器中，然后垂直寄存器列组又以一次一行的方式将信号电荷转移到水平移位寄存器内，而水平寄存器再逐位把这些电荷输出。

两种输出方法相比较，帧传输的灵敏度较高，但器件芯片面积较大。行间传输空间频率响应较好，但其灵敏度较低，而且工艺实现较困难。

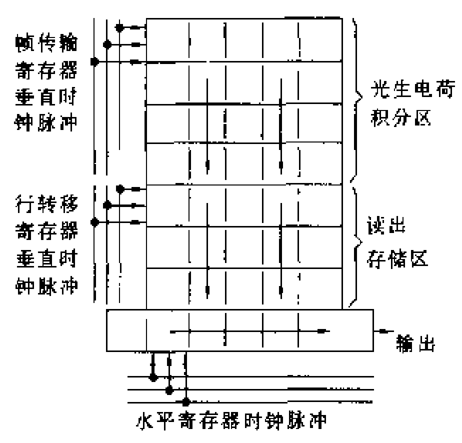


图 2-17　CCD 帧传输方式

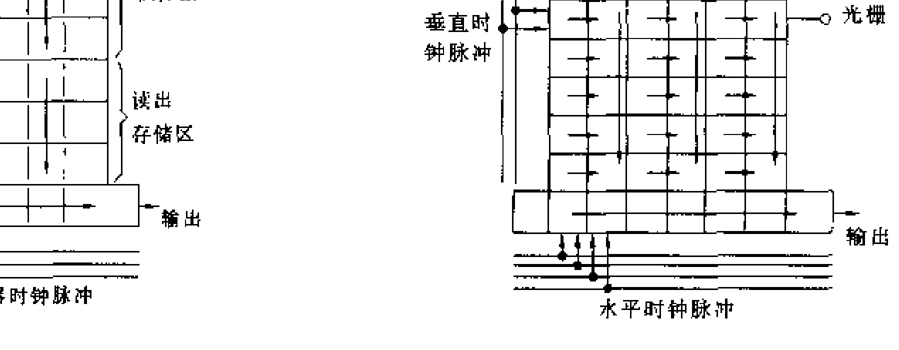


图 2-18　CCD 行间传输方式

线阵摄像器件的像素是一维阵列，因此，主要用于数码识别和传真等场合。

RCA SID52501 固体摄像器件的图像传输方式为帧传输；像素数为 512×320；像素尺寸为 0.03mm×0.03mm，灵敏度为 3250μA/1m，饱和曝光为 2.67×10^{-2}lx · s，峰-峰信号电流为 250nA，视频带宽为 3MHz，水平时钟速率为 6.1MHz，传递速率为 0.28MHz，光积分时间为 16.67ms，水平极限分辨率为 240TVL/H，垂直极限分辨率为 480TVL/H，γ 系数为 1，成像面暗电流 4nA，信噪比为 50dB。

固体摄像器件在许多领域得到广泛应用。由于 CCD 摄像器上每个光生电荷包的位置都可精确测出，因此，它拍摄的图像定位精度很高，已在天文观测上得到应用。

④ 彩色图像传感器

利用上述摄像器件可构成单色图像传感器，也可以做成彩色传感器。彩色图像传感器分为三管彩色摄像机和单管彩色摄像机。三管彩色摄像机的结构框图如图 2-19 所示。由图可见，三管彩色摄像机是由镜头、滤色镜、分色棱镜、三个摄像管(R、G、B)、放大器、扫描发生器、寻像器及电源等附属设备组成。

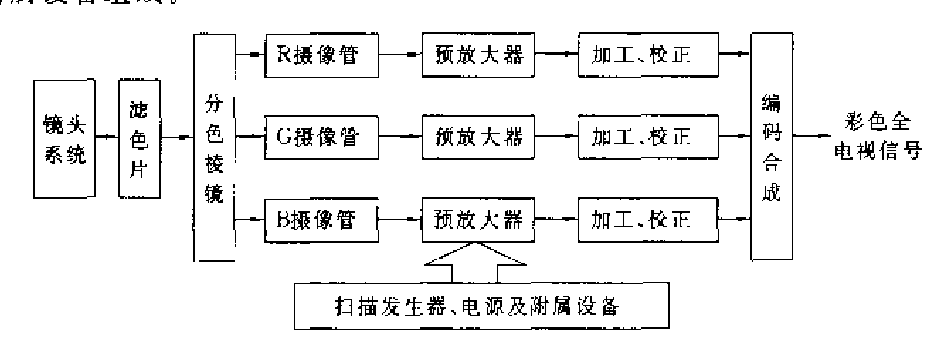


图 2-19　三管彩色摄像机结构框图

摄像机摄取彩色图像或景物，首先通过中性滤光片和色温滤光片，然后进入分色棱镜分解为三基色光像，三基色光像分别送入三个摄像管转变为图像电信号。然后，三路信号分别经过校正(电缆校正、黑斑校正、轮廓校正、彩色校正、γ 校正及电平调节等)，最后送入编码器以形成彩色全电视信号。

单管摄像机只有一个摄像管，它没有分色棱镜，是利用管内直接镀在靶上的条纹滤色器实现 R、G、B 分色作用的。单管彩色摄像机又分为频率分离式、相位分离式和三电极式三种。

频率分离式摄像机的结构框图如图2-20所示。在摄像管靶面设置两片条状滤色器，一片为反红条状滤色器，一片为反蓝条状滤色器，当光学图像经过滤色器和摄像管转变为电信号时，绿色信号是时间的连续函数，而红色和蓝色所对应的信号为时间采样函数。这些电信号经低通滤波器和带通滤波器即可分出亮度信号、红色信号及蓝色信号，经编码后就形成彩色全电视信号。

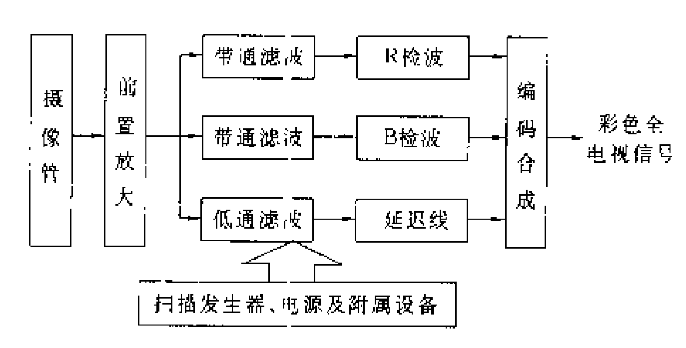


图2-20　频率分离式单管摄像机框图

相位分离式单管式摄像机是在靶前设置带状透明导电膜和红、绿、蓝垂直条纹滤色器。当电子束扫描时会得到一组组红、绿、蓝点顺序信号，通过低通及带通滤波器后可得到相应的频谱信号。这种摄像机的白平衡好，稳定性高，但清晰度不高。

三电极式摄像机是在三电极摄像管内装有重复的红、绿、蓝垂直条纹滤色器，并分别对准各自的相互绝缘的条状透明电极。当电子束扫描时，便把靶面上形成的图像或景物的三个基色的光像转变为三个基色信号，并分别通过各自的电极汇总到三条母线上，于是输出R、G、B信号，经编码后形成彩色全电视信号。这种摄像机白平衡、彩色重现和稳定性均好，但有时由于条状电极的静电电容影响，容易串色，影响清晰度。

2. 飞点扫描设备（FPS）

飞点扫描器是另一种常用的图像信息输入设备。它是由飞点扫描管、偏转系统、透镜、漫反射球及电子倍增器组成，其结构如图2-21所示。飞点管是一种亮度高、聚焦好的小型显像管，在这里它是光源。工作时，飞点管发出的光点照射到输入图像上。每一瞬间只照射一个像素，被照亮的像素反射的光射到光电倍增管上转换成电信号。飞点管的电子束在偏转线圈的作用下进行扫描，因此，光点也就在图像上扫描，最终将一幅光学图像转换为电视信号送出。彩色飞点扫描器与单色飞点扫描器的工作原理完全相同，只不过它有三个带有红、绿、蓝滤色镜的光电倍增管。它们分别把三种反射光变成电信号，从而得到R、G、B信号，这三种信号经编码即可输出彩色全电视信号。飞点扫描器设备简单，工作稳定可靠，扫描速度高，控制方式灵活。缺点是设备笨重，只能摄取照片和底片，不能输入实景。飞点扫描器也可以作为图像输出设备输出硬拷贝。

3. 鼓形扫描器

鼓形扫描器又叫光电滚筒扫描器。它的结构框图如图2-22所示。它是由滚筒、光源、反射镜、光电倍增器及机械传动系统组成。鼓形扫描器主要用来输入或输出图片。它的工作过程如下：首先把待输入图像照片覆盖在滚筒上，光源的光通过反射镜照射到图像上，图像的反射光经过透镜及光孔照射到光电倍增器上。由光电倍增器管将光信号转变为电信号。为了把图像分解为像素信号，并把这个信号按顺序送出去，滚筒在转动的过程中还要作水平移动或者滚筒转动，光源作水平移动。这样，光束便对图像作顺序扫描。图像是有灰度层次的，当光束照射到

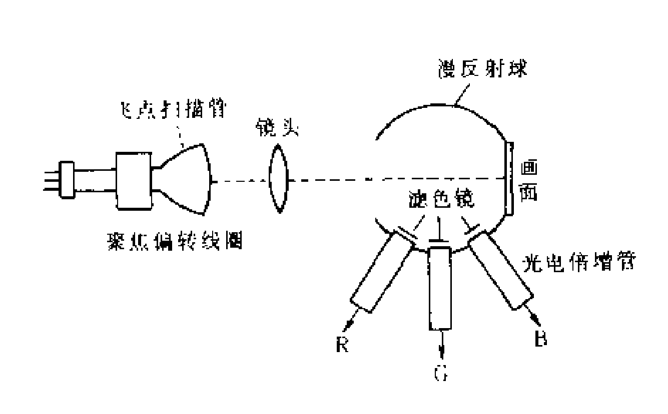


图 2-21 彩色飞点扫描器结构示意图

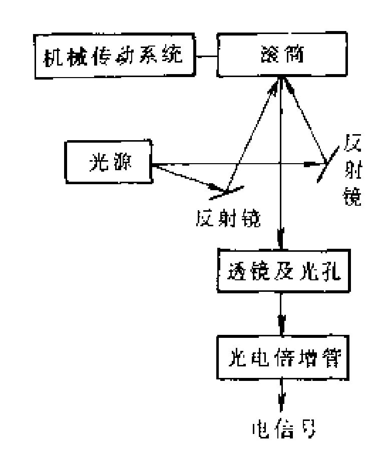


图 2-22 鼓形扫描器原理图

不同灰度位置时，其反射光的强弱也发生相应的变化。因此，光电倍增管的输出电信号的强弱也会发生相应的变化。这样就把一幅光学图像转化为电信号。这种设备既可以扫描照片、文件资料等不透明的图像，也可以扫描透明的底片、X 射线底片等图像。它既可以作输入设备用也可以作输出设备用。在作输出设备用时，只要将感光胶片放在滚筒上，反过来用图像信号控制光点强弱，使胶片逐点感光，就可以得到一幅输出图像。鼓形扫描器的结构简单、精度高。它的主要缺点是速度慢、价格较贵，为了保证其精度，要求有高质量的维护与操作。

4. 其他图像输入设备

除前述的几种图像信息输入设备外，尚有光敏二极管矩阵图像信息传感器，激光扫描器和图像位置检出器等一些输入设备。这些图像信息转换和输入设备的性能比较列入表 2-3 中，以供参考。

表 2-3 图像信息输入设备性能比较表

图像输入设备		机械式	视像管	析像管	CCD	飞点扫描器	激光扫描器	光敏二极管
性能特点	清晰度	高	中	中	中/高	高	高	中/高
	线 线	好	差	好	中等/好	好	好	好
	速 度	慢	快	快	快	快	快	中等
	信噪比	高	中等	低	中等	高	高	中等
	存储效应	无	大	无	小	小	无	小
	适用波长	中	长	中	中	长	长	中
	扫描方式	顺序	顺序	顺序	顺序	顺序	顺序	顺序
	其他	位置精度较高，光密度精度高，扫描速度慢，价格昂贵	位置及光密度较差，结构简单，使用方便，灵敏度高，扫描度快，价格便宜	灵敏度与分辨率不能兼顾，寿命长扫描度快，价格便宜	体积小、重量轻，功耗低可靠性好，几何失真小	使用灵活，但不能在亮环境中应用，扫描速度快	干扰小，方向性强，单色性好，速度高、清晰度高，精度高	比较经济

2.2.2 图像处理中的输出设备

图像处理系统输出设备的主要任务有二个：一个是将处理前后的图像显示出来以供分析、识别和解译之用；另一个是以硬拷贝或以数据的形式记录下来永久保存。图像处理系统中可采用的输出设备是多种多样的，例如前面介绍过的飞点扫描器、鼓形扫描器、激光扫描器等均可输出图像硬拷贝，而磁带机，磁盘机、光盘以及磁鼓等可用来存储图像数据。此外，监视器、打印机等也是图像处理系统中常用的设备。

图像输出显示的对象可以是文字、数字和符号；可以是单色或彩色的图形、图表等；也可以是一般图像，其中包括静止的或活动的图像，单色或彩色图像乃至立体图像。

对图像显示所要求的功能和性能可归纳为如下几个方面：①显示功能 包括能够显示的像素数目、显示的灰度层次、显示的颜色种类和范围。此外还有显示精度(定位精度和图像畸变)、显示速度、有无记忆功能及与其他图像重叠显示功能等等；②图像质量 包括画面尺寸、画面形状、显示形态、亮度、对比度、分解力、清晰度、信噪比、图形畸变、电光变换特性、余辉及闪烁等等；③其他有关性能 包括操作、维护及调整的难易，稳定性，寿命、功耗乃至体积大小等等。

下面介绍几种常用的图像输出设备。

1. 监视器

监视器是图像处理系统中必不可少的图像输出显示设备。它的关键部件是显像管。这种被称作布老恩管的阴极射线管是1897年提出来的。自那时以来虽经过了近四分之三个世纪，但管子形态并没有根本的变化，所作的大量的工作也只是缩短管子长度、减小功耗、提高质量、降低成本等方面的工作。

(1) 显像管

显像管(阴极射线管)的基本结构如图2-23所示，它由电子枪、电子束偏转系统和荧光屏组成。这些部件的主要功能如下：电子枪用来发射电子，并使之形成加速和聚焦的电子束，根据输入信号的大小，可控制电子束的强弱；偏转系统使电子束作水平或垂直偏转，以便使电子束根据输入的要求打在荧光屏的指定位置；荧光屏随着入射电子束的强度发出不同强弱的光，从而显示出可供观看的图像。

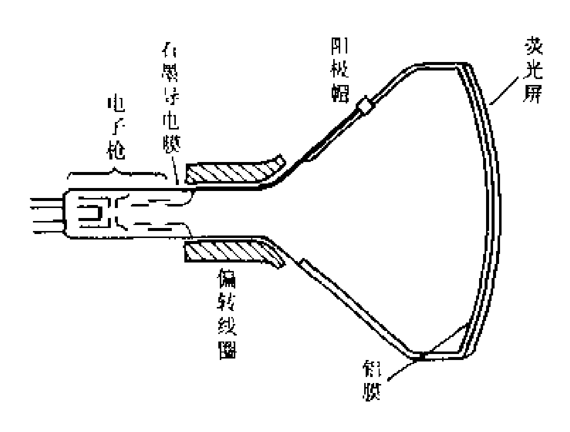


图2-23 显像管的基本结构

目前，图像处理系统中的监视器多为彩色监视器。当然，彩色监视器的心脏是彩色显像管。彩色显像管有下面几种类型。

① 荫罩式彩色显像管

这种管子是由红、绿、蓝信号调制的三条电子束穿过荫罩板上的小孔，然后各自去激发相应的荧光粉点(一般红、绿、蓝三色点呈三角形排列)，使它们发光而显示出彩色图像。这种管子工作方式简单，能可靠地工作，生产较容易。但是它的效率较低，80%～85%的电子束被荫罩板挡住了。另外，会聚调整较复杂。

② 单枪三束式显像管

单枪三束式显像管是日本索尼公司1968年研制的一种彩色显像管。这种显像管荧光屏上的荧光粉不是呈圆点状而是呈条状排列。每三条(R、G、B)成一组垂直排列。荧光屏约有500多组这种的三色条。这种单枪三束管的选色板叫做影条板，上面开有500多条细缝。电子束即从细缝通过打在荧光条上。这种管子的荧光屏和影条板只能做成柱面形。另外一种单枪三束管的选色板是栅槽式的，它是影条板的改进型，选色板和荧光屏可做成球面形。在这种显像管中，三条电子束共用一个电子枪，聚焦透镜的直径可以做得较大，因此能实现更好地聚焦。单枪三束管的优点是会聚简单。由于三条电子束水平排列，并且绿束位于中心轴上，因此，绿光栅不需会聚校正。红、蓝电子束对称排列于绿束两边，所以，只要作左右会聚校正就可以了。另一优点是清晰度好。由于聚焦好，电子束直径小且密度大，故清晰度好，同时也提高了图像的亮度。此外，这种类型的管子受地磁场影响小，电子透过率也可达20%。

除以上两显像管外，还有三电子束控色栅管、单电子束控色栅管等等。以上介绍的是几种常用的彩色显像管。用阴极射线管构成的监视器是图像处理系统中应用最为广泛的输出设备。随着技术的发展，这种显像管的清晰度和亮度正在逐步提高。

(2) 监视器

监视器的种类很多，根据使用目的不同，其电路结构也有很大区别。在广播电视中为了解决兼容与传输方面的问题发明了三种彩色制式，即：NTSC (National Television Systems Committee) 制；PAL (Phase Alternation Line) 制和 SECAM (Sequential Couleur à Mèmoire) 制。制式不同的监视器电路结构略有不同，它们只能用于显示与自己制式相同的视频信号，不能混用。

① PAL 制监视器

PAL 制的监视器原理框图如图2-24所示。它的工作原理如下：当彩色全电视信号送入视频输入端后，经带通放大器选出色度信号，进而由梳状滤波器分离出色度信号分量，然后，通过同步检波分别得到色差信号。解调副载波的频率和相位由色同步信号锁定，在识别出PAL倒相行后，由电子开关实现这种处理。其中，自动色度控制(ACC)电路用于稳定色度信号电平。

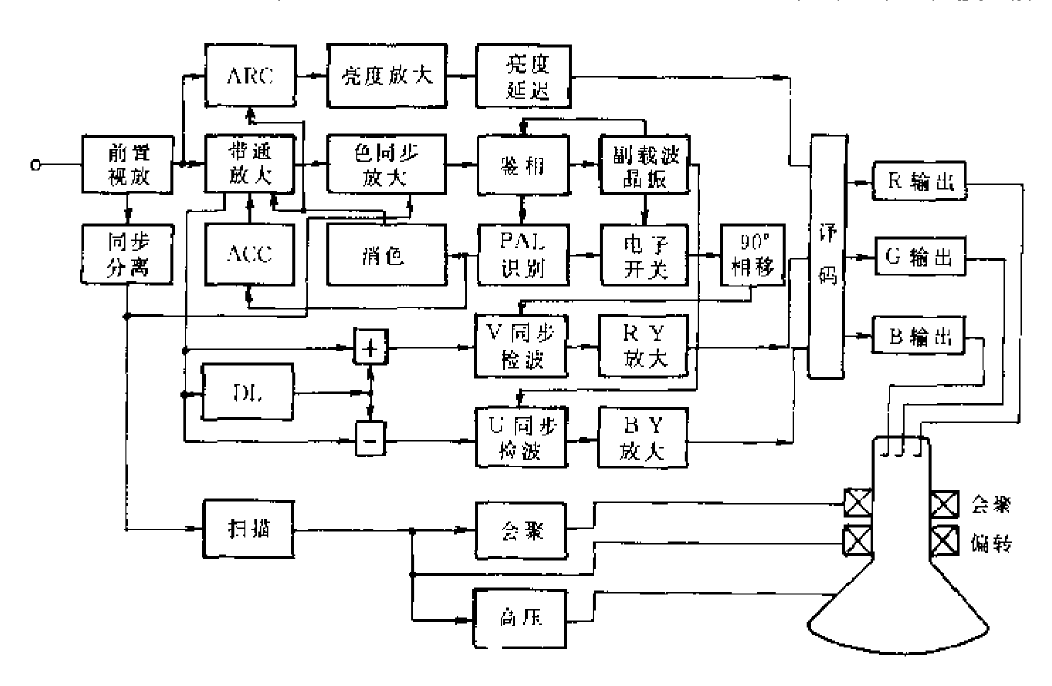


图2-24 PAL制彩色监视器原理框图

消色电路的作用是在接收到黑白视频信号时关闭色通道，以减少色杂波。彩色全电视信号的另一路送入亮度通道，为抑制色度信号的干扰，先通过副载波吸收网络，然后经放大、延迟与色差信号一起送入译码网，以便译出R、G、B信号送入显像管。另外通过同步分离电路分离出同步信号以控制扫描电路，以便在荧光屏上显示稳定的图像。这种监视器可用于广播电视系统，也可以用于图像处理系统。

② R、G、B分别输入的彩色监视器

这种监视器原理框图如图2-25所示。监视器的电路十分简单，红、绿、蓝三个图像信号分别送入三路通道，经放大后直接激励显像管的R、G、B三个阴极。另外，复合同步信号送入同步通道，经同步分离分别控制行扫描和场扫描以便得到稳定的图像显示。这种监视器只能用于图像处理系统，不能用于广播电视系统。

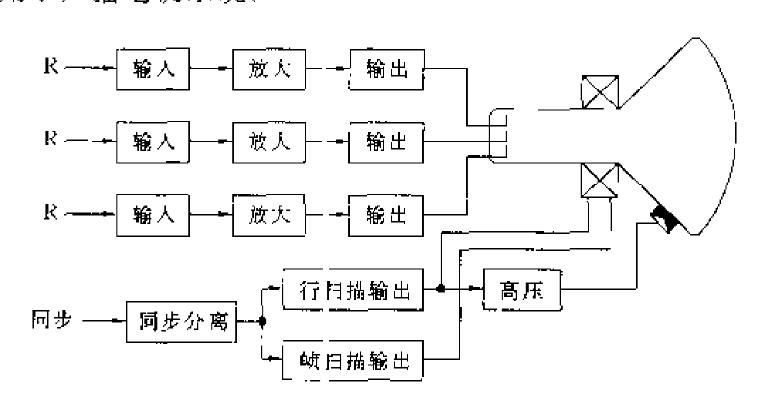


图2-25 R、G、B分别输入的彩色监视器原理框图

以CD351为例，其指标如下：屏幕尺寸为20英寸，分解力为1024×768像素，放大器带宽为20MHz，上升和下降沿小于或等于25ns，振铃与上冲小于10%，水平扫描为15.5kHz到23.5kHz三种，垂直为50Hz、60Hz、72Hz、80Hz四种，几何失真最大为画面高度的1.5%，R、G、B三信号输入幅度为$0.7V_{p-p}$，同步信号为$4V_{p-p}$，具有亮度、对比度控制及手动消磁控制。

2. 激光扫描器

激光扫描器可以用作输入设备也可以用作输出设备。它是用光束代替电子束。光束由激光器产生，激光束方向性强、单色性好、亮度高，因而图像清晰。

激光扫描是由两个多面镜鼓实现的。多面镜鼓是一种圆柱体，圆柱体的侧面切成很多镜面，镜面上镀有高反射率的介质膜或金属膜，使之成为反射性能极好的镜面。多面镜鼓的扫描运动如图2-26所示。扫描器由一个水平扫描镜鼓和一个垂直扫描镜鼓组成。激光束首先射向水平扫描镜鼓的一个镜面a_1，然后反射光束再射向垂直扫描镜鼓一个镜面b_1，经b_1反射投向屏幕。在扫描中，两个镜鼓分别以不同的速度转动。由于光源是固定的，水平镜面鼓的转动，使得光束的入射角度从小到大变化，因此，反射光线也在作同样的变化（反射角等于入射角），使投向屏幕的光束随着反射角的变化作从左到右的扫描运动。当镜面a_1转过去后，随之而来的是镜面a_2、a_3、…，它们分别重复a_1的过程，在更换镜面的瞬

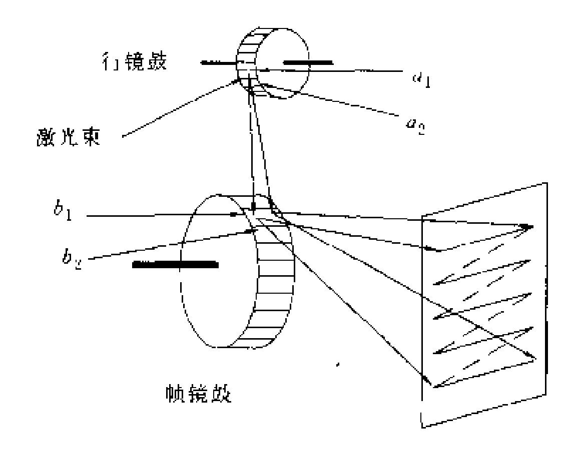


图2-26 多面镜鼓的扫描运动

间便是行扫描的回扫。垂直扫描镜鼓的扫描过程与水平扫描道理相同,只不过它的转动较慢。当行镜鼓转 n 个面时,垂直镜鼓转动一个面,则在屏幕上就会扫出一个具有 n 行的光栅。如果用图像信号去调制激光束的强弱,那么在屏幕上就会扫出一幅图像来。在屏幕位置放上感光相纸,就可以制成图像的硬拷机。

3. 其他图像显示装置

除了前述的两种图像输出装置外,还有一些其他图像显示设备,限于篇幅,这里不可能详细讨论它们的工作原理,在此仅开列一些装置的类型以供参考。

(1) 彩色打印机

彩色打印机可被计算机控制直接打印出彩色图像。

(2) 投影显示器

投影显示器多用于大屏幕显示。可分为有源显示和无源显示。投影管显示为有源显示类型,而光阀显示和激光显示等为无源显示类型。这主要看光源本身是否有调制功能而定。

(3) 平板显示

平板显示包括气体放电平板显示、液晶显示、发光二极管显示、电致发光显示等等。平板显示现在已在许多场合得到应用,但其显示性能有些尚难与阴极射线管相比拟,因此,有人预言:在未来相当一段时间内,图像显示的主流设备仍然是阴极射线管,待其价格、性能、尺寸、重量等方面发展到优于显像管时,方可以代替显像管。概括地说,平板显示设备有下面几种。

① 液晶显示器(LCD)

液晶的发现已有100多年的历史,但真正用于显示技术的历史不到30年。尽管LCD要想取代CRT至少还需15年左右的时间,但其发展势头之大,发展速度之快却令人刮目相看。LCD的突出性能是极吸引人的。它的体积小、重量轻、低电压、微功耗、无软X射线,几乎可以做到与CRT相媲美的全色显示和相当的亮度。目前的主要问题在于观察视角、分辨率和屏幕尺寸等技术指标还不如CRT。但最近的最新产品已大有改善,如Sharp公司推出的彩色非晶硅TFT-LCD,屏幕尺寸21英寸,分辨率640×480,像素数921600点,彩色数1670万。富士通10.4英寸显示器的视角可达120°。

目前,平板液晶显示器的种类有如下几种:

STN-LCD (Super Twisted Nematic)(超级双扭式向列相);

TN-LCD (Twisted Nematic)(双扭式向列相);

TFT-LCD (Thin film Transistor)(薄片晶体管式);

CSH-LCD (Color Super Home-ostrophic)(彩色超级同向扭曲)。

② 场致发光显示器(FED)

场致发光平面显示器有多种。总体来说,从技术上看还不能与CRT和LCD相竞争。但等离子显示器件的性能优于LCD。其本身视角可达160°,且结构工艺简单,目前是有力的竞争者。1994年已有40英寸的壁挂式AD-PDP显示器展出。目前,场致发光显示器的主要问题是亮度低、寻址线路复杂、功耗大。另外电致发光显示(ELD)是全固态的,视角宽、响应快,但亮度低、全色显示困难。至于彩色荧光显示目前只能用于字符显示。

场致发光显示(FED)具有光明的前途。FED是最新发展起来的彩色平板显示器件。SID95'会议上展出的产品6英寸方形的全色FED分辨率达到512×512。FED有以下优点:

a) 薄型,无需加背光源,比LCD还薄;

b) 易于拼接,可做成大屏幕显示器;

c）可以高度集成制成高清晰度和高亮度显示板；

d）电压低、功耗小、寿命长，如 TFT-LCD，背光源功耗为 13W，而相同尺寸的 FED 只有 5W；

e）图像质量好。FED 在高对比度下可获得 1600 万种颜色和 256 级灰度，分辨率为 1024×768，无视角限制，响应速度快（≤2μs）；

f）发光亮度起伏小，成品率高；

g）不需偏转线圈，无软 X 射线，抗辐射和电磁干扰，工作温度宽（－40～＋85℃）；

h）生产工艺简单，而且 FED 每个像素有数十个至数百个阴极，即使几个不工作也没有影响。

但目前要解决的问题是大面积的 FED 要改善发光的均匀性和提高低压荧光粉的发光效率，在实用化中封接、排气、真空维持等工艺有困难，这些问题解决后 FED 大有前途。

2.2.3 数字图像处理的主机系统

如前所述，数字图像处理系统主要由主机和输入输出系统组成，近年来围绕着数据量大和处理速度这一矛盾，发展了许多图像处理系统。下面介绍几种常见的图像处理系统及其性能。

1. 国产图像处理系统

图 2-27 是我国研制的一种图像处理系统框图。这个系统以国产 130 计算机为核心，加上一些外围设备组成。输入设备由摄像机和数字化器组成；输出设备为黑白监视器和彩色监视器；此外还有磁盘、磁带、打印机及字符 CRT。

一幅图像经摄像机摄入后，通过数字化器变为数字信号。然后送入图像存储器。这个存储器兼有输入及输出显示两个任务。在将图像数据送入计算机之前，首先经 D/A 变换送入监视器，监视一下输入的图像是否理想，如果不理想可重新输入，如果满意，则可通过总线送入计算机存入原始图像。反过来经处理后的图像通过总线系统送入存储器，此时，存储器作为刷新显示之用，以便显示处理结果。这套系统的精度为 256×256×8bit，输入速率为 1.5μs/pixel，输出速度为 2μs/pixel，RAM 读写时间为 600ns，通道传输时间为 1.8μs/pixel。这是一个结构简单，价格便宜的通用处理系统。

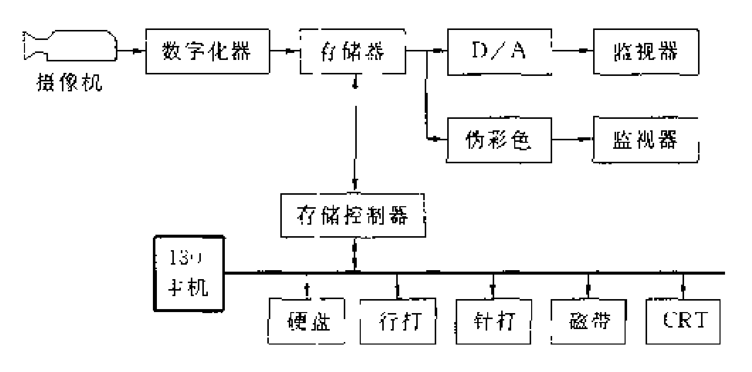


图 2-27 图像处理系统框图

2. Model 75 图像处理系统

美国 FS 公司的 Model 75 图像处理系统的结构如图 2-28 所示。这套系统由主机子系统、视频输入子系统，图像处理器，视频输出子系统及交互式控制子系统组成。

主机子系统可由下列机型组成：VAX-11、PDP-11、HP3000 系列 Ⅱ 或 Ⅳ、Motorola 68000 等。主机子系统可配接磁带机、硬盘、打印机及多个终端。

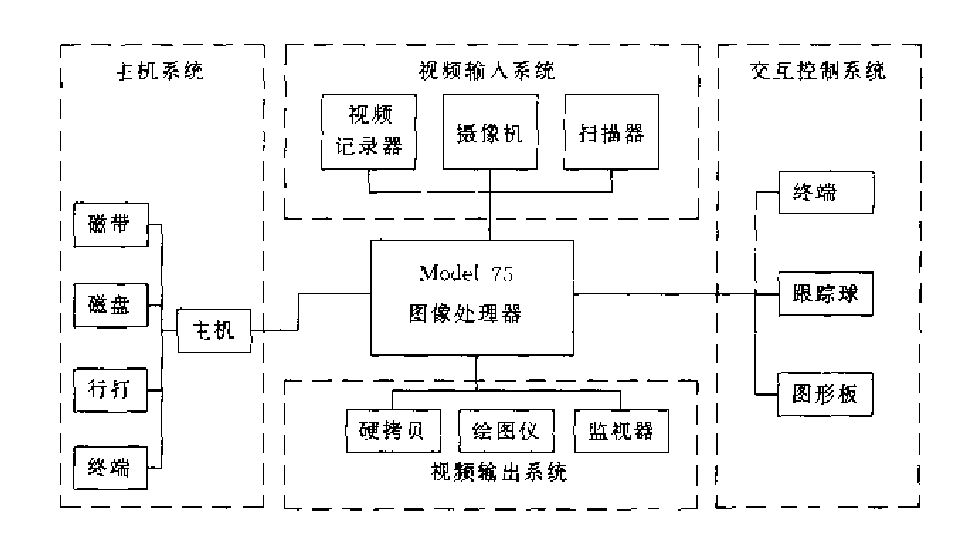


图 2-28 Model 75 图像处理系统

视频输入系统可以是摄像机、飞点扫描器及视频记录器。视频输出系统由监视器，绘图器及硬拷贝器组成。图像处理部分为 Model 75 图像处理器。

(1) 系统的软件

主机系统的系统软件为 RSX-11M、VMS、MPE-1V 或 UNIX。Model 75 子系统配有多任务磁盘操作系统 System 575。System 575 是一个分层的、模块式的软件包，这些软件有如下一些内容：

1) Model 7540 处理软件包，它提供了硬件显示接口程序。

2) Model 7541 接口和应用程序包，它包括 30 个 Fortran 子程序和 30 个接口子程序，用来控制 Model 75 的各种子系统，如刷新、存储、查表及光标等。

3) Model 7542 是诊断软件包。

4) Model 7543 是源程序包，它有 60 个 Fortran 程序，可作加、减、乘、除、平均，编码、2D-FFT、滤波、均衡、查表、放大等等常用处理。

(2) 系统主要指标

输入为标准视频信号，视频输出 625 行、25 帧、视频幅度 $1U_{p\text{-}p}$，刷新存储器为 512×512×8bit，图形平面为 512×512×1bit。硬件放大系统有×2、×4、×8 等，3 路并行 12bit 实时管道处理，屏幕显示可作 512^2，256^2、128^2，64^2 等开窗。

3. PC-2000 图像处理系统

PERICOLOR2000 (简称 PC-2000)是法国纳梅克公司生产的交互式图像处理系统。系统结构如图 2-29 所示。

(1) 硬件结构

PC-2000 的硬件由如下一些部分构成：

1) 存储器

① 512×512×8bit 图像存储器块(I_1 到 I_8)；

② 512×512×8bit 工作存储器块(T_1,T_2)；

③ 512×512×4bit 光标存储器块；

④ 128KB PROM；

⑤ (256÷512)KB RAM；

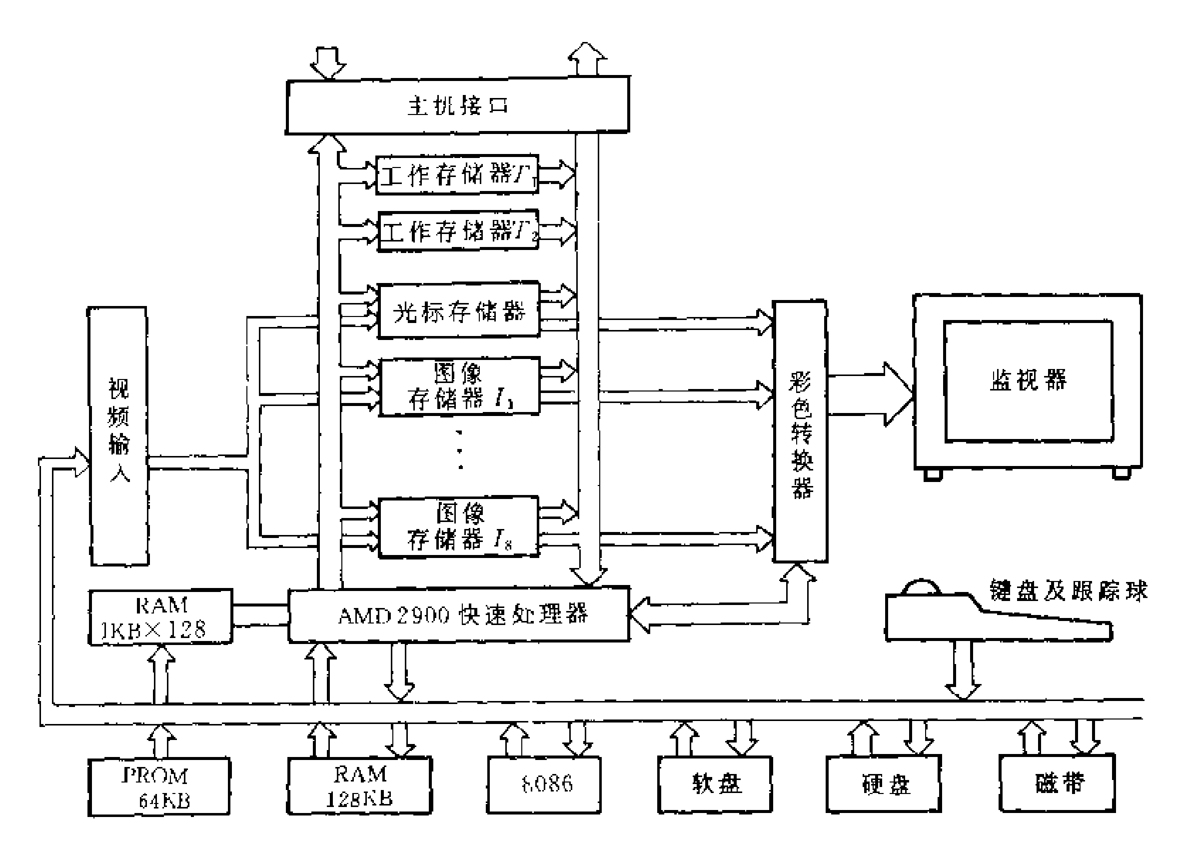


图 2-29　PC-2000 图像处理系统框图

⑥ 128×1KB RAM。

2）微处理器

系统配有16bit的Intel8086微处理器，主要用于管理；另外，配有128bit的AMD 2900位片式快速处理器，主要用于快速处理。

3）接口

① 16bit面向主计算机的输入输出接口；

② V24（RS232）串行输入输出接口；

③ 24bit并行输入输出接口，其中4bit用于跟踪球耦合；

④ 300-baud用于键盘的异步输入输出接口。

4）彩色转换

彩色转换中的函数发生器和三个8bit彩色表均系硬件操作。它的原理框图如图2-30所示。在伪彩色显示中要通过二次转换，第一次为函数转换。8个8bit图像平面经函数转换为f_1～f_8，经求和电路求得$\sum R = f_1 + f_2 + \cdots + f_8$，同理求得$\sum G$和$\sum B$。第二步将$\sum R$、$\sum G$和$\sum B$分别送入$R_t$、$G_t$和$B_t$三个彩色表进行第二次转换，转换后三路分别送到D/A变换器转换为红、绿、蓝视频模拟信号送入监视器显示。彩色选择总数可达2^{24}（16×10^6种）彩色组合。除伪彩色显示外，还有真彩色显示功能。

5）彩色监视器

系统配有高分辨率彩色监视器，屏幕尺寸为51cm，R、G、B及同步分别输入方式。

6）软盘驱动器

双面双密度 8 英寸软件驱动器二个，每个 1000kB。

7）50MB 温氏硬盘

8）快速图像数字化器

标准电视摄像机，四通道 A/D 转换，转换一幅 512×512×8bit 图像 20ms。

9）磁带机 800/1600BPI

10）激光硬拷贝机

11）图形板

12）彩色喷墨打印机

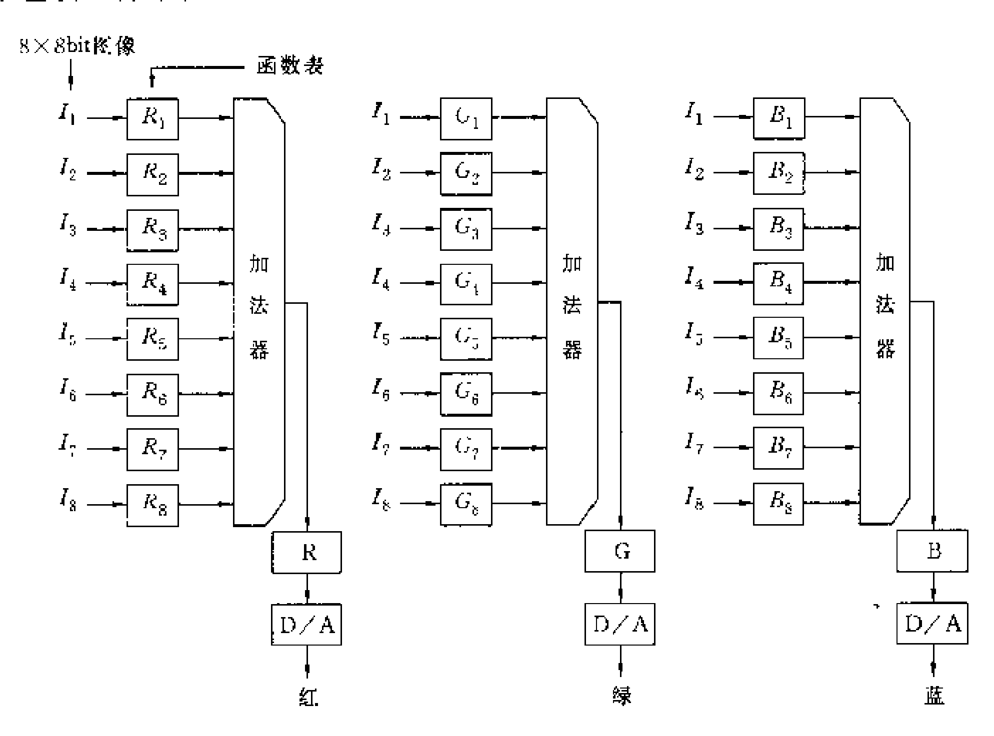


图 2-30　数字/彩色转换框图

（2）软件

处理系统的系统软件为 RMX86 操作系统，其中有各种编辑、链接及库文件等软件。此外，配有 Fortran86、PLM86 等高级语言及汇编语言。

系统中有大量的常用的图像处理程序，它们均以“菜单”的形式实现人机对话，为用户提供了极方便的应用条件。PC-2000 系统功能均列于“主菜单”中，并用一个字符代表各“子菜单”只要按压相应的键就可实现一种处理功能。例如：“I”代表图像子菜单，在这个子菜单下可实现对图像的放大、开窗、移位等处理；“F”代表伪彩色子菜单，按压功能键 F，就可调出伪彩色配置菜单；“T”代表处理子菜单，按压 T 键，可调出如下一些标准的图像处理功能，这些功能可在 2 幅图像 P、Q 及操作者定义的参数 K 间实现。例如：

=：复制 P=Q 或 P=K

I：交换 P↔Q

+：加法 P=P+Q 或 P=P+K

-：减法 P=P-Q 或 P=P-K

*：乘法 P=P*Q 或 P=P*K

⋮

<：图像压缩

>：图像扩展

等等。此外尚有动画处理功能，可按功能键 A 实现，这个功能在医学上观察心脏跳动及实现心脏图像处理是很有用的。

(3) 性能特点

本系统有较多优点，它既可以独立使用也可以与主计算机相连作为一个工作站使用；它的“菜单”式的交互方式为用户提供了极大的方便等等。由于采用了高速处理器实现图像处理，因此，处理速度较快，例如：传送一幅 512×512×8bit 的图像到存储器只要 200ms，对 512×512×8bit 图像作 3×3 卷积只要 3ms，取一幅图像进计算机只要 20ms 等等。在作伪彩色、函数变换等近于实时处理。除此之外，系统配有较齐全的外设，可适应各种不同的需要。除常规处理软件外，系统还有多种应用软件可供选用，这为用户提供了很多方便。

2.3 视觉系统

在图像处理中所采用的许多处理技术，其主要目的是帮助观察者理解和分析图像中的某些内容。因此，图像处理系统不但应该是从视觉系统的角度来看是理想的系统，而且又是最经济有效的系统。为了达到预期的目的，在图像处理中不但要考虑图像的客观性质而且也要考虑视觉系统的主观性质。本节将对视觉系统的基本构造及其特性作一些讨论。

2.3.1 视觉系统的基本构造

人的视觉系统是由眼球、神经系统及大脑的视觉中枢构成。人的眼球的横断面如图 2-31 所示。人眼的形状为一球形，其平均直径约 20mm。这球形之外壳有三层薄膜，最外层是角膜和巩膜。角膜是硬而透明的组织，它覆盖在眼睛的前表面。巩膜与角膜连在一起，它是一层不透明的膜，包围着眼球剩余的部分。巩膜的里面是脉络膜，这层膜有血管网，它是眼睛的重要滋养源。脉络膜外壳着色很深，因此，有利于减少进入眼内的外来光和光在眼球内的反射。脉络膜的前边被分为睫状体和虹膜。虹膜的收缩和扩张控制着允许进入眼内的光量。虹膜的中间开口处是瞳孔，瞳孔的大小是可变的，大约可以从 2mm 变到 8mm。虹膜的前部有眼睛的明显的色素，而后部则含有黑色素。眼睛最里层的膜是视网膜，它布满了整个后部的内壁上。当眼球被适当地聚焦时，从眼睛外部物体来的光就在视网膜上成像。晶状体由纤维细胞的同心层组成，并由睫状体上的睫状小带支撑着。它含有 60%～70%的水，约 60%的脂肪。晶状体被稍黄的色素染色，其颜色随年龄的增长而有所加深。它吸收可见光谱的 8%，波长越短吸收的越多。红外光和紫外光被晶状结构内的蛋白质大大地加以吸收。但是，过量的红外线和紫外线会伤害眼睛。除此之外，还有把光刺激传给大脑的神经系统及保护眼睛的眼睑和泪腺等附属组织。

视网膜可看成是大脑分化出来的一部分，它的构造比其他感觉器官都要复杂，它具有高度的信息处理机能。其结构模型如图 2-32 所示。视网膜的厚度大约为 0.1～0.5mm。参与信息处理的细胞有视觉细胞(包括锥状体和杆状体)、水平细胞(Horizontal cell)、埃玛克里细胞(Amacrine cell)、两极细胞(Bipolar cell)和神经节细胞(Ganglion cell)等 5 种。眼睛中的光接收器主要是视觉细胞，它包括锥状体和杆状体。在图 2-32 中所示的中央凹(或称中心窝)部分特别薄，这部分没有杆状体，只密集地分布锥状体。锥状体只有在光线明亮的情况下才起作用，具有辨别光波波长的能力，因此，对颜色十分敏感。有时它被叫做白昼视觉。每只眼睛的锥状体

大约有七百万个，在中央凹的分布间隔大约为 2～2.5μm。杆状体比锥状体的灵敏度高，在较暗的光线下就能起作用。但是，它没有辨别颜色的能力，有时又叫它夜视觉。杆状体分布在视网膜表面上，分布面积较大，其数量大约有一亿三千万个。正因为两种视觉细胞的不同特点，所以我们看到的物体在白天有鲜明的色彩，而在夜里却看不到颜色。与视觉细胞相比，神经节细胞数目较少，大约有一百万左右。

锥状体和两极细胞的关系及细胞的结合方式并不十分明显，一般在数量上，在中央凹约为 1∶1 左右。这些细胞接受光刺激后，通过神经系统传入大脑的视觉中枢，同时也可以控制眼球的转动，使感兴趣的物体的像落到视网膜的中央凹上。

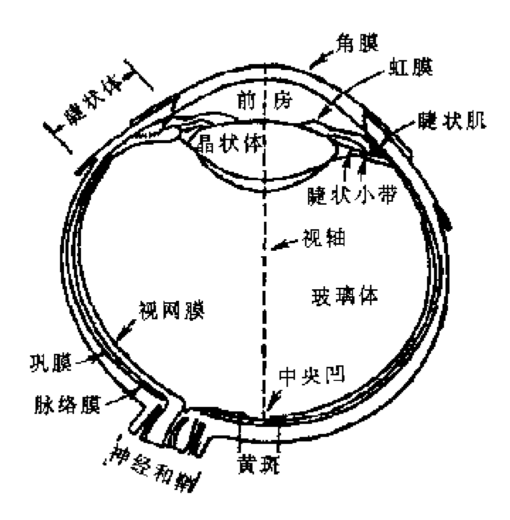


图 2-31　人眼的断面图

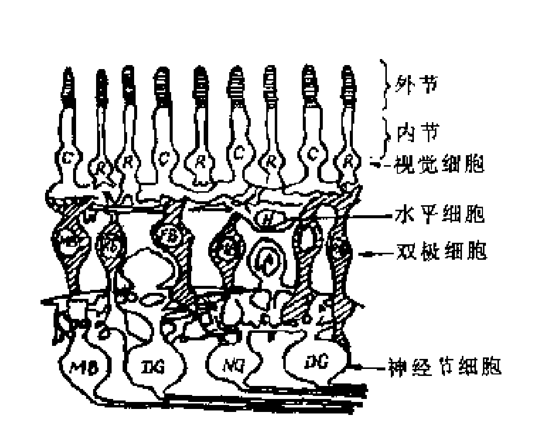


图 2-32　视网膜结构模型

2.3.2　光觉和色觉

眼睛对光的感觉称为光觉，对颜色的感觉称为色觉。这是眼睛的基本特性。

1. 光觉门限及亮度辨别门限

产生光感觉必须有一定量的光进入眼睛，把产生光觉的最小亮度叫做光觉门限或光觉阈。光觉门限的适应状态受生理条件、光的波长、光刺激的持续时间、刺激面积以及在视网膜上的位置等因素的影响。光觉门限值大约为 1×10^{-6}cd/m²(尼特)。人眼感觉光的范围的最大值和最小值之比达到 10^{10}以上。

锥状体和杆状体各自的最大灵敏度随波长而异，其标准灵敏度特性如图 2-33 所示。其中 a 代表锥状体灵敏度曲线，b 代表杆状体灵敏度曲线。由图可见，杆状体的最灵敏点比锥状体最灵敏点波长短 50nm 左右。波长从 380～740nm 分别与紫、蓝、绿、黄、橙、红等顺序相对应。这就是在傍晚光线变暗时我们所看到的物体没有颜色的原因，这种现象叫 Purkinje shift 现象。

光觉门限与刺激面积和刺激时间有密切关系。关于光觉门限与刺激面积的关系有里克(Ricco)定律和里波(Riper)定律来描述。当刺激面积较小时，等于光觉门限的强度 I 与面积 A 的关系遵循下式之关系，即：

$$IA = 常数 \tag{2-47}$$

这条定律就叫做里克定律。这里面积 A 的大小在如下范围内：在中央凹处，视角在几分之内；离中央凹 4°～7°处的亚中央凹处大约为 0.5°以内；距中央凹处 35°左右约 2°以内都适合本定理。当刺激面积较大时，有式(2-48)的关系成立：

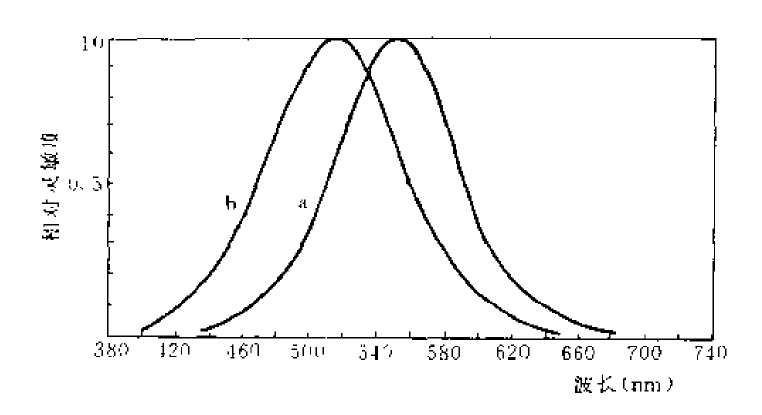


图 2-33　锥状体和杆状体的相对灵敏度特性

$$I\sqrt{A} = \text{常数} \tag{2-48}$$

这个关系就是里波定律。这个定律在视角范围内都可成立。一般情况下，里克定律和里波定律常合在一起用 IA＝常数来表示，统称为里波定理。

关于光觉门限与时间的关系由布洛克(Block)定理来描述，它在时间较短的范围内才成立。布洛克定理是指光强等于光觉门限时，刺激时间 T 与光强度 I 的关系如式(2-49)所示：

$$IT = \text{常数} \tag{2-49}$$

式(2-19)约在 0.1s 以下的范围内成立。

光觉门限是指产生光觉的最小值，而辨别门限是指辨别亮度差别而必须的光强度差的最小值。这个最小值 ΔI 随光强 I 的大小而异。有时也采用相对辨别门限 $\Delta I/I$ 或称之为韦伯比来表示辨别门限。

亮度的相对辨别门限 $\Delta I/I$ 与光强度水平 I 及刺激面积的关系曲线如图 2-34 所示。这些曲线是斯坦哈特(Steinhardt)在 1936 年测定的。

由图可见，开始时随 I 的增大，$\Delta I/I$ 减小，当 I 增大到一定值后，$\Delta I/I$ 则稳定在某一值上不再变化。在 $\lg I=0$ 处有一个不平滑点，这是因为处在杆状体和锥状体交替起作用的强度处。

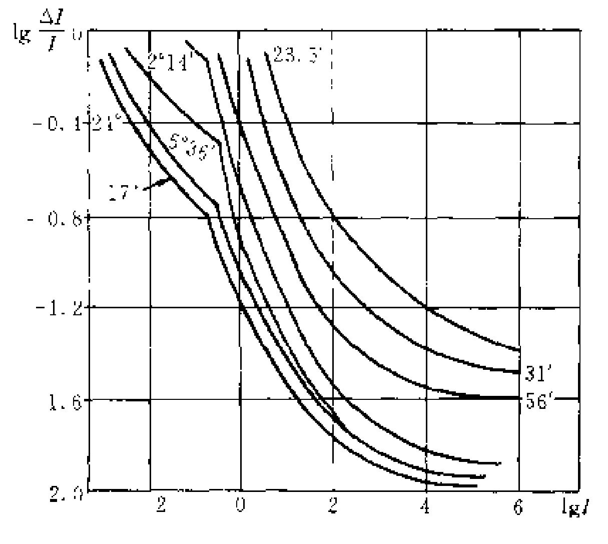


图 2-34　相对亮度辨别门限 $\Delta I/I$ 与光强 I 及刺激面积 A 的关系曲线

亮度辨别门限与光觉门限一样受刺激时间和面积的影响。刺激时间 T 在某一范围内 ΔI 值与 T 成反比。ΔI 与刺激面积的关系有与皮埃隆(Pieron)定理相似的关系成立。

关于光的波长与辨别门限的关系由赫克特(Hecht)在 450～670nm 进行了测定。一般规律是波长越长则辨别门限越高(即辨别灵敏度低)。

在中央凹处及其周围的辨别门限也进行了测定，一般周围的辨别门限大，特别是在明亮的场合下，这种倾向将增强。

2. 有关色觉的学说

自1730年牛顿成功地分解了太阳光谱以来，认为光的波动经过神经传到大脑，由于波长不同而产生不同颜色感觉的这一假说至今还在提倡。最初，Young(1801年)认为颜色不是光的物理性质而是一种感觉现象。后来赫姆霍尔兹(Helmholtz)发展了这种假说，认为视网膜有三种色细胞，由于光学反应引起三种视神经纤维的兴奋，由此又引起大脑三种神经细胞兴奋而产生色觉。这就是扬-赫姆霍尔兹(Young-Helmholtz)的色觉三原色学说。另一方面，也有对三原色假说持反对立场的人，特别是扬-赫姆霍尔兹的三原色为红、绿、紫。据经验，黄色用红色和绿色混合而成是难于理解的。也就是说，用他的三原色假说不能说明黄色的纯色性。提出这一反对论点的代表人物是赫林(Hering)。赫林的假说是在视网膜上有红-绿物质，黄-蓝物质，白-黑物质。在光刺激下，各物质同时向对立的方向发生化学变化，向两个方向变化的程度根据刺激的波长不同而不同，由此产生色觉。赫林的相对色假说对说明色适应和色对比现象理由较好，但对色盲的性质却不能加以详细说明。三原色假说和相对色假说考虑方法是对立的，本世纪也有提倡一种折衷的假说，如Ladd-Franklindel发展假说，Hart-Ridgedel多色假说等等。最近采用物理手段研究生理学的方法发展很迅速。

3. 色觉的生理学结构

关于人眼的锥状体感光色素和色觉关系的研究由拉什顿(Rushton)在1957年建立了良好的开端。他收集到入射到眼睛中的光在眼底的反射成份，分析其波长，结果显示出在正常人的中央凹锥状体中存在着叫做红敏素(Erythrolabe)(吸收最大波长$\lambda_{max}=590nm$)和绿敏素(Chlorolabe)($\lambda_{max}=540nm$)的两种感光色素，特别是预言了叫做青蓝(Cyanolabe)色素的存在。并且指出红色色盲者不能检出红敏素色素来，这样，对色盲和视物之间的关系问题给出了有力的启示。经过这一研究证明锥状体内至少有两种感光色素，它们存在于同一锥状体内。马克斯(Marks)在1964年用显微分光法对动物和人的单一锥状体内感光色素进行了测定。证明了每个锥状体内具有简单的视觉物质存在。

2.4 光度学及色度学原理

亮度和颜色是进入眼睛的可见光的强弱及波长成份的一种感觉的属性。从某一入射光产生的亮度和颜色的感觉无法测定，并且这种因人而异的感觉也不能比较。既使对同一个人来说，由于观察条件不同感觉也不一样。既然这些感觉无法测定那么用什么方法来表示亮度和颜色呢？下面对光度学和色度学以及视觉特征做一些讨论。

2.4.1 颜色的表示方法及观察条件

颜色的表示方法大体上有二套方法。一种是设置一套作为标准的颜色样本，被试的颜色与样本进行比较，然后用特殊的记号来表示。具有代表性的例子就是芒塞尔(Munsell)表示系统。另一种方法是取决于刺激光的物理性质和色的感觉的对应关系。用与这个规定相对应的量来表示试料的光的物理性质。这就是国际上规定的CIE表示系统。在处理光的物理性质和色觉的关系时，可在单纯的条件下来决定这种规定。一般条件规定如下：

1) 刺激亮度在视觉细胞的锥状体起作用又不刺眼的范围内；

2) 观察视野在2°(或10°)范围内，而且范围外是黑暗的；

3) 视野内的光分布均匀并且不随时间变化。

在这样单纯化了的条件下观察颜色时，记录刺激光的物理性质具有一贯性。但是，在这种

条件下观察到的颜色与日常在复杂情况下观察到的颜色不一样，为与感觉色相区别，把这种颜色叫做心理物理色。

2.4.2 三基色混色及色度表示原理

根据光的波动说，单一波长的光称为单色光。从人们可以区别各种不同颜色这样一个事实出发，似乎可以假设视网膜上也存在着许多不同类型的锥状体，每一类型的锥状体只“谐振”于某一特定的颜色。如果锥状体果真有这样的单色响应，那么某一彩色感觉只能由相应波长的电磁能引起。然而，事实却不完全如此，照射到视网膜上的某一单色光并不是引起该彩色的惟一因素。例如，有几种单色黄光可以由射到视网膜上的红光和绿光配出来。几乎所有的彩色都能由三种基本彩色混配出来，这三种彩色就叫做三基色。

由三基色混配各种颜色的方法通常有两种，这就是相加混色和相减混色。彩色电视机上的颜色是通过相加混色产生的，而彩色电影和幻灯片等与绘画原料一样是通过相减混色产生各种颜色的。相加混色和相减混色的主要区别表现在以下三个方面。第一，相加混色是由发光体发出的光相加而产生各种颜色，而相减混色是先有白色光，尔后从中减去某些成份(吸收)得到各种彩色。第二，相加混色的三基色是红、绿、蓝，而相减混色的三基色是黄、青、紫(一般不确切地说成是黄、蓝、红)。也就是说相加混色的补色就是相减混色的基色。第三，相加混色和相减混色有不同规律(指颜料相混)，这些将在后续章节中详细叙述。

著名的格拉斯曼定律反映了视觉对颜色的反应取决于红绿蓝三输入量的代数和这一事实。格拉斯曼定律包括如下四项内容：

1) 所有颜色都可以用互相独立的三基色混合得到；

2) 假如三基色的混合比相等，则色调和色饱和度也相等；

3) 任意两种颜色相混合产生的新颜色与采用三基色分别合成这两种颜色的各自成份混合起来得到的结果相等；

4) 混合色的光亮度是原来各分量光亮度的总和。

这里色调、色饱和度及亮度是表示色觉程度的。色调是表示各种颜色的种类的术语，色饱和度表示颜色深浅。以三基色为基础的格拉斯曼定律可用下式表示：

$$F = R(R) + G(G) + B(B) \tag{2-50}$$

2.4.3 CIE 的 R、G、B 颜色表示系统

国际照明委员会(CIE)选择红色($\lambda=700.00$nm)、绿色($\lambda=546.1$nm)和蓝色($\lambda=435.8$nm)三种单色光作为表色系统的三基色，这就是 CIE 的 R、G、B 颜色表示系统。

在数字图像处理中的终端显示通常用显像管(CRT)也就是用彩色监视器显示。由相加混色原理可知白光(W)可由红(R)、绿(G)、蓝(B)三种基色光相加得到。产生 1lm 的白光所需要的三基色的近似值可用下面的亮度方程来表示：

$$1\text{lm}(W) = 0.30\text{lm}(R) + 0.59\text{lm}(G) + 0.11\text{lm}(B) \tag{2-51}$$

由式(2-51)可见，产生白光时三基色的比例关系是不等的，这显然给实际使用带来一些不方便。为了克服这一缺点，使用了三基色单位制，这就是所谓的 T 单位制。在使用 T 单位制时，认为白光是由等量的三基色组成。因此，式(2-51)表示的亮度方程可改写如下：

$$1\text{lm}(W) = 1\text{T}(R) + 1\text{T}(G) + 1\text{T}(B) \tag{2-52}$$

比较式(2-51)和式(2-52)可以看出：1T 单位红光 = 0.30lm；1T 单位绿光 = 0.59lm；1T 单位

蓝光＝0.11lm。由此可知T单位与流明数的关系，在需要的时候可以很容易地进行转换。由于T单位的采用就除掉了复杂数字带来的麻烦。

由三基色原理可知，任何颜色都可由三基色混配而得到。为了简单又方便地描绘出各种彩色与三基色的关系，采用了彩色三角形与色度图的表示方法。

彩色三角形以最简单的直观图形给出三个给定彩色相混合时所获得的彩色的大致范围。彩色三角形如图2-35所示。这是一个等边三角形，三个顶点分别为红、绿、蓝三色。其中黄色位于红与绿之中间，紫色落在蓝色和红色中间，青色在绿色与蓝色中间。三角形的中心就是三基色分量都相等的白色。以三角形三个边为界，其内部每一点都是三基色混合而成的颜色。穿过中心点的任何一条直线所连系的两种彩色互为补色，即两者混配起来就形成白色。越靠近三角形中点则色饱和度越低。

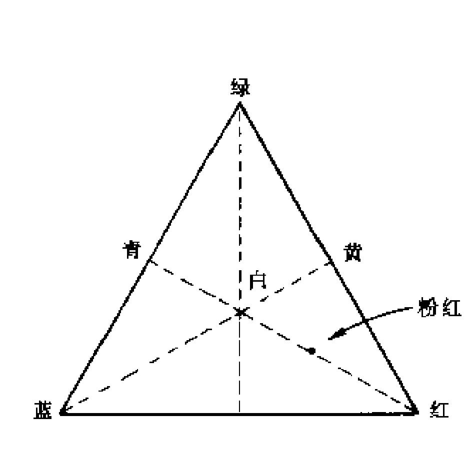


图2-35　相加混色彩色三角形

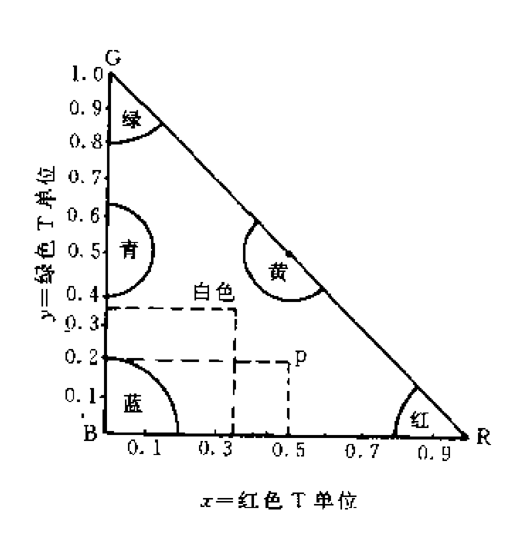


图2-36　直角彩色三角形

对彩色的感觉必须考虑三个量，即色调、色饱和度和亮度。彩色三角形是二维图形，因此，它只能表示色调和色饱和度，不能表示亮度。

一般在彩色三角形上标上标度就可直接读出产生某给定色调所需的三基色比例，为此，采用直角三角形更为方便。图2-36便是采用直角三角形表示的彩色三角形。图中每一色调的量取为1个T单位。红色沿着x轴，绿色沿着y轴，给定的蓝色可简单地用格拉斯曼定律推出。因为每一色调都规定1个T单位，所以，任一色调的红、绿、蓝分量总和为1，蓝色的T单位数就可以从1减去红色和绿色的T单位而推出。例如图2-36的P点，这一色调位于$R=0.5$、$G=0.2$处，这说明，1T单位的"P"色调包含有0.5T单位的红色和0.2T单位的绿色。由格拉斯曼定律可算出蓝色量为$1-(0.5+0.2)=0.3$T单位。也就是说1T单位的"P"色调等于0.5T单位的红色，0.2T单位的绿色，0.3T单位的蓝色之和。

上述彩色三角形并不是在任何时候都适用，这主要是他们只能表示出特定的三基色所能获得的色调。

在CIE色度图中，采用了假想的三基色。这样就可以画出一个包括一切彩色的色度图。图2-37所示的就是CIE色度图。其中假想的三基色位于三角形的三个顶点，所有谱色都位于三角形内马蹄形曲线上。在马蹄形图的一部分曲线上注有波长数，这样就可以根据波长辨别颜色。在马蹄形的底部没有标度，这里是非谱色，波长在这里自然没有意义。所谓谱色是指能以单色出现在光谱上而且有一定波长的彩色，反之，不能作为单色出现在光谱上的颜色称为非谱

色，各种紫红色就是非谱色的例子。

图 2-37 中的C点是标准白光。作为标准光源，CIE 规定的有 A、B、C 三种，A 光源是绝对温度约为 2845K 的全辐射体发出的光，其色度坐标为 $x=0.4476$，$y=0.4075$。有代表性的就是白炽电灯发出的光。B 光源是绝对温度约为 4870K 全辐射体发出的光，其色度坐标为 $x=0.3485$，$y=0.3518$。正午时刻直射的日光与其近似。在实验室中可由 A 光源用特制的滤色镜得到。C 光源的绝对温度为 6740K，色度坐标为 $x=0.3101$，$y=0.3163$。直射的日光与天上蓝色天空的光混合起来近似于这种光源。

由于色觉与刺激视野有关，所以上述色度图对视野大约为 2°时成立。为了适应视野较大的场合，在 1964 年又定义了视野为 10°的颜色表示系统。

另外，也用“彩色图”表示颜色系统，最常见的是美国芒塞尔制。这是一个圆形的图形，分为 10 大块扇形部分，如图 2-38 所示。每一扇形又分十小块，直径两端的色调互为补色，饱和度在圆周上最大，分 16 级向圆心逐步减小，直到白色。

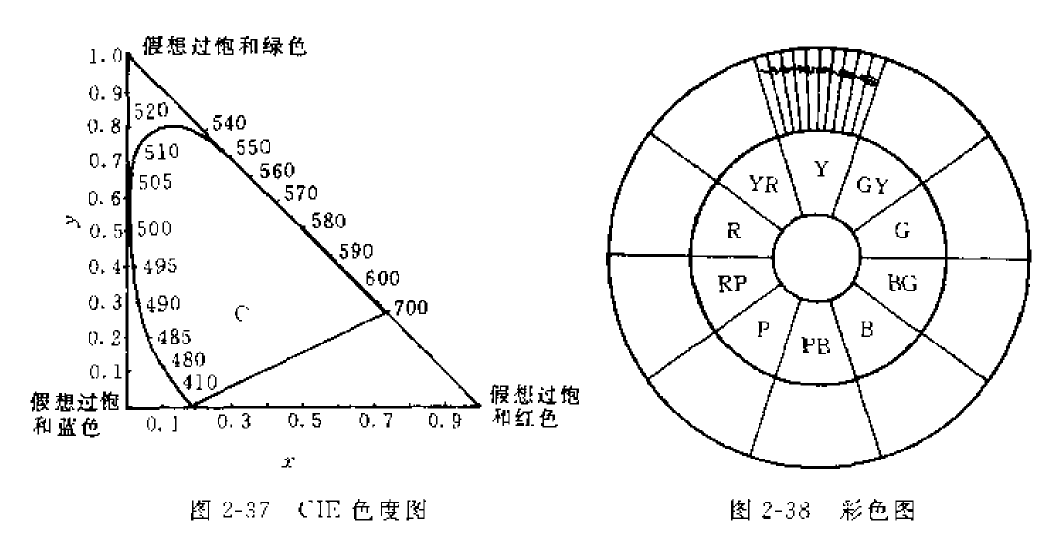


图 2-37　CIE 色度图

图 2-38　彩色图

从彩色图上可见，5Y 是最黄的颜色，10Y 则接近黄绿色，同理，5R 是最红色，10R 近于红黄色，而 1R 近于紫红色等等。总之，颜色表示系统有多种，除此之外还有 CIE1960UCS(Uniform Chromaticity Scale)、ULCS(Uniform Light Chromaticity Scale) 等。

2.5　亮度和颜色感觉的视觉特征

2.5.1　刺激强度与感觉的关系

对于感觉器官来说，刺激强度 I 产生 ΔI 的变化，而且这个变化刚刚能辨别出来，那么对应的感觉有下式成立：

$$\Delta s = k\frac{\Delta I}{I} \tag{2-53}$$

两边积分后即成为式(2-54)的形式：

$$s = k\lg\frac{I}{I_1} \tag{2-54}$$

这个式子说明感觉量与刺激强度的对数成比例，式中的 I_1 为绝对门限值，这个关系叫韦伯-费

克纳法则(Weber-Fechner)。

关于刺激光的强度和颜色的感觉，由于在较暗的情况下只有杆状体起作用，所以此时并没有颜色感觉。亮度达到 $10^{-3}cd/m^2$ 时锥状体才起作用，也就是亮度达到所谓微明视觉的水平时才有色觉。颜色刺激与颜色感觉的对应关系比较复杂，光的辉度变化时，颜色也变化，这种现象叫贝佐尔得-布鲁克(Bezold-Brucke)现象。

2.5.2 亮度适应和颜色适应

从较亮的场所到较暗的场所时，很难马上看到东西，相反，从较暗的场所到较亮的场所，也看不见东西。一般把眼睛的状态适应明暗条件的变化叫亮度适应。从亮到暗的变化叫暗适应；从暗向亮的变化叫亮适应。一般亮适应时间较短，暗适应较长。与亮度适应相区别，随着光的分布而变化，眼睛有对颜色刺激灵敏度变化的性质。例如，用强红光刺激眼睛后看本来是黄色的物体却是绿色的。这种依赖分光分布的视觉灵敏度现象叫色适应。

2.5.3 亮度对比和颜色对比

一般情况，在相同亮度的刺激下，背景亮度不同所感觉到的明暗程度也不同。图 2-39 中的圆环的亮度是一样的，但是，由于左右两边的背景不同，看到的圆环两边的明暗程度也不一样，背景暗会觉得圆环亮一些；背景亮会感觉圆环较暗。在观察颜色的场合也一样，在图形的色度一样，但背景颜色不一样时，感觉到的图形的色度也不一样。

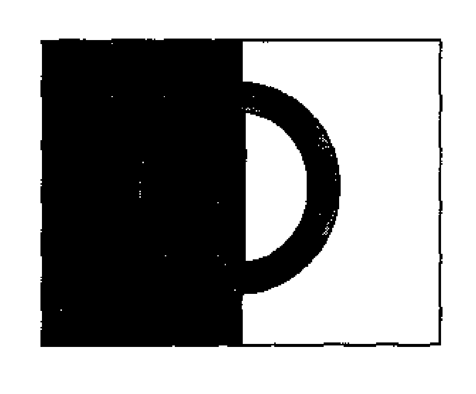


图 2-39　明暗对比的例子

刺激的亮度和色度受周围背景的影响而使其产生不同感觉的现象叫同时对比现象。这里包括亮度对比和颜色对比。另外，在二个刺激相继出现的场合，后续刺激的感觉受到先行刺激的影响，这种现象叫相继对比。但是，相继对比可以看作是视觉的时间特性或适应效果的一个侧面，因此，一般的对比多指同时对比。实验表明，在背景亮度比目标亮度低的场合，感觉目标有一定亮度。当背景亮度比目标亮时，看到的目标就有暗得多的感觉。同时对比效果在背景大的场合比较显著，但不一定在目标被包围的情况下才产生，在其他场合也可以产生，只是效果小一些罢了。

关于对比效果有一定性的法则，即基尔希曼(Kirschman)法则。其基本内容如下：

1) 目标比背景小，颜色对比大；

2) 颜色对比在空间分离的两个领域内也发生，间隔大时则效果较小；

3) 背景大，对比量亦大；

4) 明暗对比最小时，颜色对比最大；

5) 明暗相同时，背景色度高对比量大；

6) 亮度及颜色的恒定性。

有这样一些例子可以说明亮度的恒定性。例如，我们感觉一张白纸的亮度，照明光的强度改变时它也不怎么改变，总有一定的亮度感觉。更为显著的例子是，与白天的煤山相比尽管夜间的雪山亮度低，但我们感觉煤山还是黑的，而雪山还是白的。这种物理亮度在变化而感觉却有保持一定的倾向叫亮度的恒定性。

另外，在照明光的颜色稍微改变的场合，我们感觉白纸仍然是白纸。这一类照明光改变但

感觉到物体颜色能稍微保持一定的倾向叫颜色的恒定性。这些恒定性与亮度和颜色的适应性与对比因素有关，同时也与材质有关。

2.5.4 颜色感觉与刺激面积的关系

对于一个色觉正常的人来说，颜色刺激面积非常小时就不能识别颜色了，这种色觉异常状态叫第三色盲。米德尔顿(Middleton)和霍姆斯(Holmes)曾用视角为2′和1′的目标研究了色度变化的感觉。他们得到的结果是当面积较小时，橙、蓝、绿直线的感觉很接近，当更小时，就只有灰色的感觉了。这个原理被用到彩色电视机原理中，对减少传送频带作出了贡献。

2.5.5 主观颜色

如图2-40所示的黑白圆板。当它旋转时，我们将看到色度较低的颜色。因为这个现象不能用刺激光的分光特性来预测，所以叫主观色。有时也叫Feahner color。当模型板缓慢旋转时，会看到a、b、c、d位置上的不同颜色。

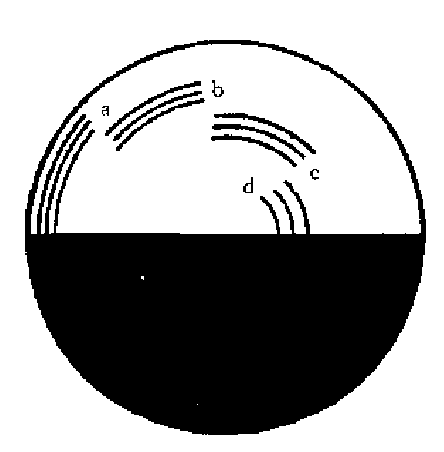


图2-10 贝纳姆(Benham)模型

2.5.6 记忆色

我们经常接触的物体，对它特有的颜色会有记忆，这就是记忆色的概念。例如人的皮肤颜色、草木的颜色等很容易从颜色样本中选出来，这是人人都有的视觉倾向。但是，一般来说记忆色与实际颜色并不一样，记忆色的色度和亮度比实际颜色都要高。记忆色与理想颜色的再现关系密切，无论在彩色电视技术上还是绘画上都受到广泛的关注。

2.5.7 进入色、后退色、膨胀色、收缩色

我们观察物体时，根据其颜色有的感觉距离较近，有的感觉较远。使你有拉近距离的颜色叫进入色，它有较长的波长。看着有推远距离感觉的颜色叫后退色，它有较短的波长。另外，有的颜色，使我们有物体变大的感觉，这叫膨胀色。有的会有缩小的感觉，这叫收缩色。这些特性又与亮度有关。亮度高的黄色有增大的感觉，亮度低的蓝色有缩小的感觉，除此之外，还有所谓的暖色、冷色等等。

2.5.8 颜色和爱好

颜色与人的感情有关。人们也各自有自己所爱好的颜色。这种爱好会随着性别、年龄、时代等因素而变化。另外，两种颜色相邻时，由配色而产生的和谐感也是一个重要的问题。有人作了大量的研究，提出了一些所谓彩色和谐理论。

2.6 视觉的空间性质

2.6.1 视力

视力是指人眼分辨物体细微部分的能力。眼睛的视野是较广的，一般以视线为中心向鼻

子一侧大约为 65°,向耳朵一侧约为 100°～104°,向上约 65°,向下约 75°这样一个范围。在这样宽的视野范围内,视力最好的那一部分仅仅在视线附近,也就是只在视网膜中央凹上。在中央凹周围视力则急剧下降。根据不同的测试条件测得的视力值也不同。根据国际标准,通常采用所谓的兰多尔特(Landolt)环来测定视力值。具体测定方法如图 2-41 所示。测试条件如下:视距为 5m,照度为 500lx,环的开口缝隙则刚好使视角为 1′。当刚好勉强能分辨得开时,这种情况下的视力为 1.0。如果使缝隙为上述的 1/2 时,那么在其他条件相同时视力为 2.0。在视角范围内或者说在视网膜的各个部分上视力是不同的。在较暗的场合,因为锥状体不起作用,杆状体密度较高的中央凹附近视力相对地变高。

眼睛的光学系统对视力也有影响。瞳孔较大时,因为成像的光行差而使视力变低,瞳孔直径在 3mm 以下时,由瞳孔引起的折射和光行差互相抵消,因此,视力有一定值。但是在极端情况下,瞳孔很小时视力也要下降。

另外,影响视力的因素还有水晶体的调节状态,也就是说由于视距不同视力也不同。一般情况下,视距在 1m 以下时视距越小则视力降低越显著。此外,照明亮度高或者测试卡与背景的对比度大则视力会提高。但是,当背景用黑色,测试标为白色时,在某一对比度的场合由于光渗作用视力也会降低。特别是测试标在运动的情况下,运动速度越快则视力越低。视力与测试标运动速度的关系如图 2-42 所示。

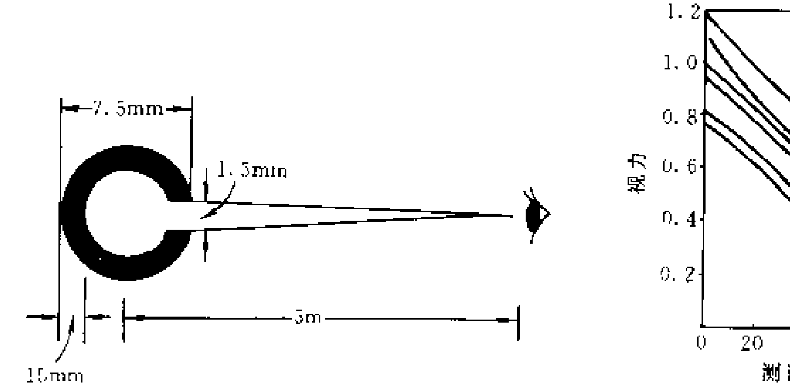


图 2-41 视力为 1.0 时的兰多尔特环

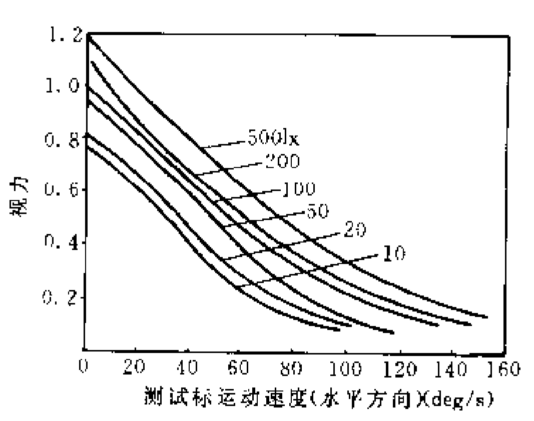


图 2-42 测试运动速度对视力的影响

2.6.2 视觉的空间频率特性

在电信号的传输系统中,常常用输入激励和输出响应之间的关系表示系统的特性。最为常用的是相位特性与频率特性等。如果系统没有非线性失真,那么响应与激励之间就有确定的对应关系,此时就可以用一个测量结果采用数学处理方法来推导其他响应。

在光学系统中,可以用空间变化的信号来代替时间变化的信号。例如空间正弦波,只要把时间量纲换成距离的量纲就可以用相同的方法加以处理。对于视觉系统,采用视力和亮度辨别门限一类参数来表示视觉系统的特性固然有重要意义,同时也希望考虑物理图像传输系统和适用于数学处理的系统这一环节。以此为基础,我们有可能采用通盘考虑视觉系统的反差灵敏度和分辨能力以及图像的主观粒状性质和清晰度等物理系统的方法进行处理和预测。

1. 眼睛光学系统的空间频率特性

眼睛光学系统的空间频率特性已由拉曼特(Lamant)、迪莫特(Demott)、克劳斯科普夫(Krauskopf)及罗尔勒(Rohler)等人进行了测定,测定结果如图 2-43 所示。图中实线是克劳斯

科普夫的测定结果；点划线是罗尔勒测定的结果；点线是迪莫特测定的结果；虚线是拉曼特测定的结果。由图可见眼球光学系统具有低通特性。

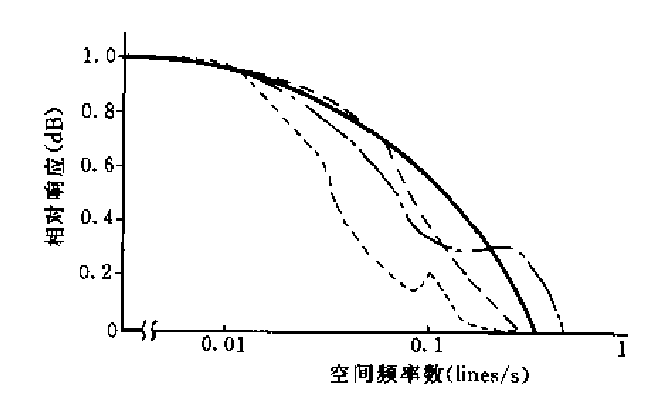


图 2-43　眼睛光学系统的空间频率特性

2. 根据主观判断求得的视觉空间频率特性

根据主观判断测定视觉空间频率特性可以这样进行，将图 2-44 所示图型(空间正弦波模

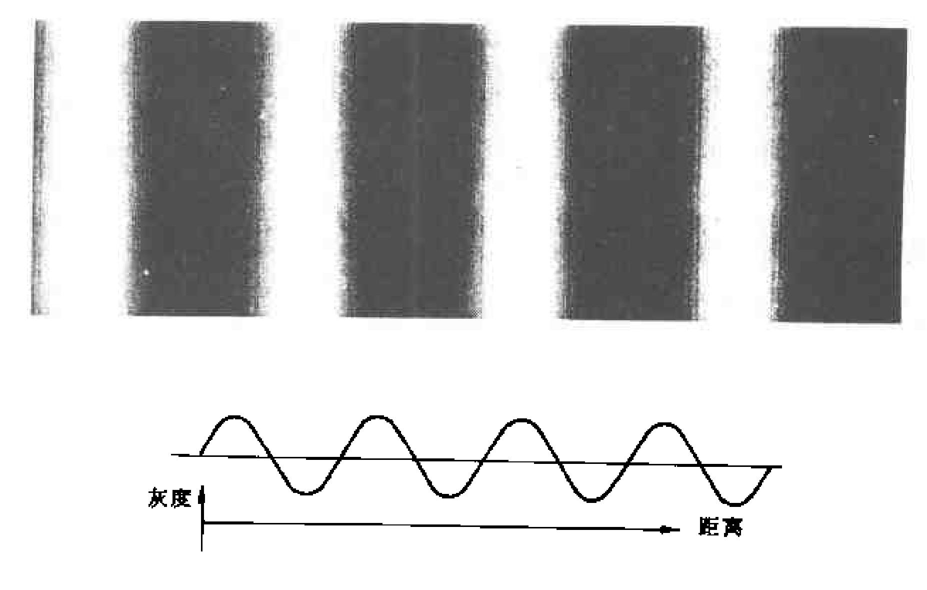


图 2 44　空间正弦波图形

型)通过光学装置或电视装置显示出来并让被试验者观看，然后使辉度固定，改变对比度，求出刚好能辨别出图形的辨别门限。或者另外准备一个标准的对比度模型，求出认为与模型相等的对比度主观等价值 PSE(Point of Subjective Equality)。根据各种空间频度数测出的曲线就表示出视觉的空间频率特性。用这种方法，很多人对上述特性进行了测试，结果如图 2-45 所示。

测试结果虽然多少有些差异，但总的倾向是一致的。在高频域和低频域其响应值均较低，其整个特性近似于带通滤波器特性。图中 A 曲线是日本 NHK 测定的，采用的平均辉度为 10 英尺 · 朗伯；B 曲线是大上(日本)的测试结果；C 曲线是谢德(Schade)在 1955 年测得的结果。视觉的空间频率特性随图像的平均辉度、提示时间的变化以及图像移动与否有关。在视觉的空间频率特性中的一个重要特性是马赫效应。马赫效应是一种轮廓增强现象。一幅明暗图像，一边亮一边暗，中间过渡是缓慢斜变的，当观看这样的图像时，视觉的感觉是亮的一边更亮，暗的一边更暗，同时靠近暗的一边的亮度比远离暗的一边要亮，而靠近亮的一边比远离亮的一边显得更暗。这就是如图 2-46 所示的上冲现象。

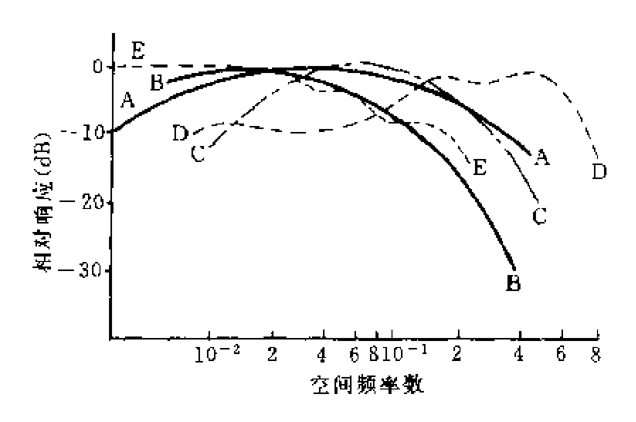


图 2-45 用对比度辨别门限测定的空间频率特性

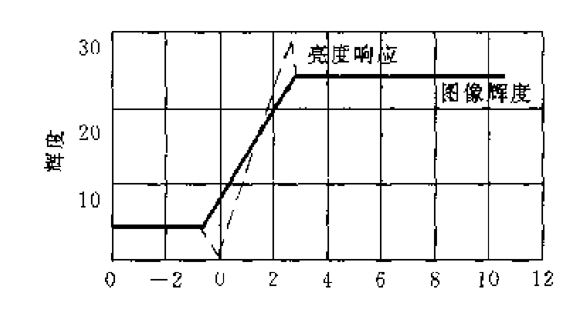


图 2-46 马赫(Mach)现象

2.6.3 颜色辨别门限的空间频率特性

谢德(Schade)和山口等人曾经用红、绿、蓝等单色空间正弦波来测定视觉的空间频率特性。实验结果表明与黑白情况几乎没有差别。用 2 种颜色组合起来的色度空间正弦波测试视觉色度空间频率特性，其结果如图 2-47 所示。图中所示是以波长为 598nm 为中心向红和绿方向变化，其特性与亮度的情况不一样，在低频范围灵敏度不下降。

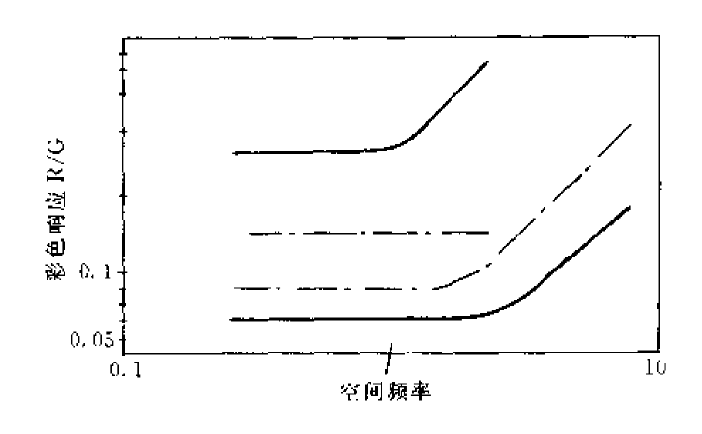


图 2-47 用辨别门限测定的色度空间频率特性

2.6.4 视觉的空间频率特性和图像的清晰度

清晰度是指图像边界的明确程度。它是表示描述细微图像能力的物理量。分解力是表示微小图像的再现能力。因此，清晰度和分解力两者尚不尽相同。50 多年前在光学领域中就引入了空间频率特性的概念，并开始研究包括视觉系统在内的图像传送的空间频率特性和主观

清晰度的关系。到目前为止提出了不少关系式，但实验结果大致上相同，但究竟用什么尺度来衡量还须进一步讨论。

就电视而言，传输频带受限时还必须考虑相位关系。单纯加大传输能量，主观清晰度不一定能提高。在NTSC制式的电视中，彩色副载波为3.58MHz，在接收机图像放大电路中要进行衰减，这样，支配清晰度的辉度信号的频带就变窄了。在这种条件下与视觉空间频率特性的峰值相当的图像频率（在距画面高度的1倍的距离观看时，这频率约1MHz）附近的增益比低频范围提高5～6dB时，图像的清晰度会显著提高。

2.7 视觉的时间特性

视觉的时间特性是指对光刺激的过渡反应特性。如对闪烁的灵敏度特性、适应性、残像、对运动的感觉及眼球运动和感觉的关系等性质。

2.7.1 加入阶跃光波刺激的明暗感觉

加入阶跃光波刺激产生的感觉如图2-48所示。由图可见，刺激后在几十毫秒时达到顶点，然后慢慢减少到一个常值。感觉曲线的上升沿随刺激光强的增加而缩短。

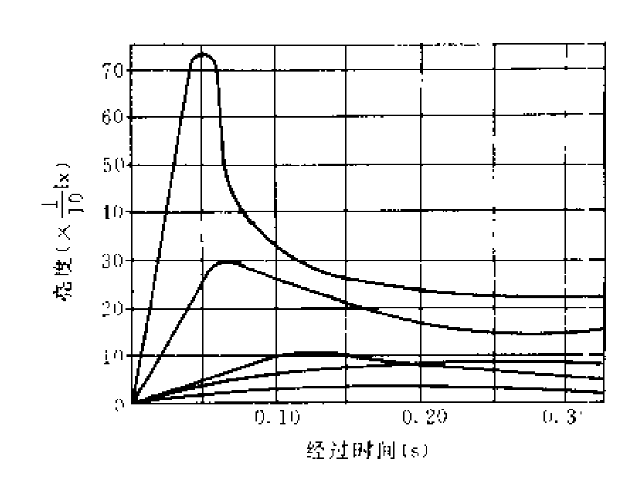


图2-48 阶跃白光刺激对明暗感觉的曲线

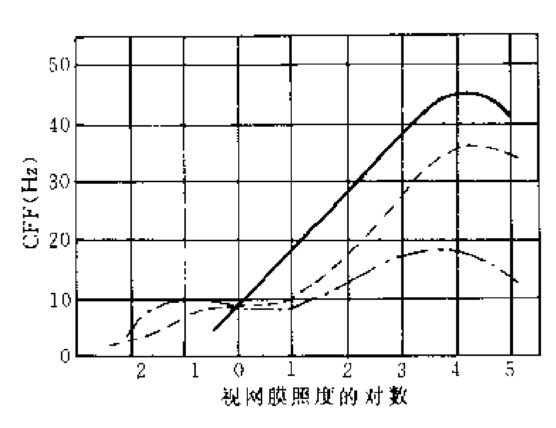


图2-49 视网膜部位及刺激光强与CFF的关系曲线

2.7.2 闪烁

当光的闪烁次数增加时，就不会有闪烁的感觉了，这时与连续光刺激的感觉一样。闪烁的次数叫临界融合频率(CFF, Critical Fusion Frequency)。CFF与照射光的强度及在视网膜上的部位不同而有显著变化。其关系曲线如图2-49所示。由图可见，离中央凹越远（偏角大）其融合频率越低。照度越高则融合频率越高。人眼锥状体的CFF值约为60～70Hz，杆状体大约是12～15Hz。CFF值与光强有式(2-55)的关系：

$$F = a\lg I + b \tag{2-55}$$

式中，F代表CFF，I表示刺激光强，a、b为常数。这个关系叫费里-波尔特(Ferry-Porter)定律。除此之外，CFF值还受背景亮度影响，背景亮度高时随着刺激光强的增大CFF值也提高。

2.7.3 视觉空间频率特性和时间因素的关系

如前所述,用空间正弦波图形测试视觉的空间频率特性,测试图形的出示时间直接影响相对对比度灵敏度。其结果如图 2-50 所示。当测试图出示时间不受限制时相对灵敏度最大,出示时间越短则越低。

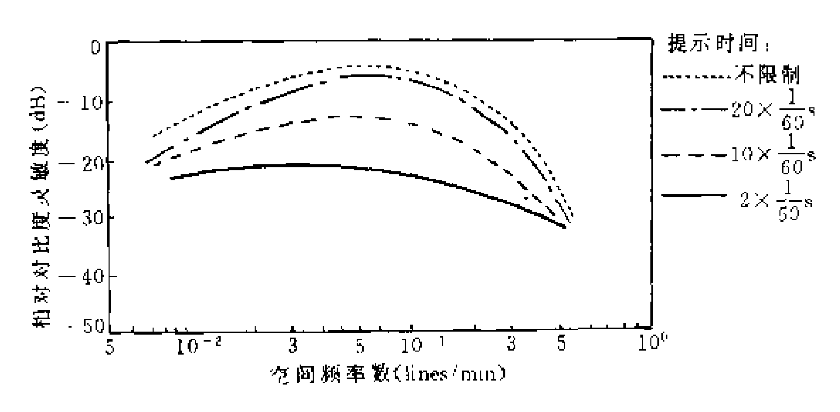


图 2-50 视觉空间频率特性与图形提示时间的关系

2.7.4 眼球运动和视觉的关系

视网膜上的视觉细胞大约有一亿多个,从眼睛出来的神经末梢不超过一百万个。黄斑处的锥状细胞大约有三万四千个,其中每个细胞都有独立的神经末梢通往大脑。黄斑以外的部分则是几个杆状体接到一个神经末梢上。因此,视力最好的部分只限于黄斑附近。在视野的大部分只能看到一个模糊的像。但是,实际上我们却总能看到一个清楚的像,这主要归功于眼球的自由运动机能。

眼球运动有断续性运动、平稳跟踪运动、辐辏开散(Vergence movement)运动和凝视运动等等。断续性运动是高速的跳跃性的运动,上升时间大约为 0.01~0.05s;平稳跟踪运动是低速眼球运动。辐辏开散运动是指两眼视线在远近方向上的运动,运动速度较低。凝视运动是指视线注视一点时常常进行类似噪声状的微小运动,它包括颤动、闪烁和偏移等形式。总之,眼睛观察物体时总是伴随着各种运动,既使是注视着某一物体也伴随着所谓凝视运动的微小运动。眼球运动有维持对比度感觉的作用。一般情况下,对于不能预测的运动要延迟 0.2s 左右才能响应,能够预测的情况下眼球运动往往先于对象物的运动。

2.7.5 运动的感觉

不仅在实际物体运动和观察者运动时有运动的感觉,而且实际上物体不运动而其他条件在运动的条件下也有运动的感觉。在心理学上把实际物体或观察者运动产生的运动感觉叫做实际运动,不是这种情况下产生的运动感觉叫假象运动、诱导运动、自动运动及运动残像等。诱导运动就是由于静止物体的周围物体运动的影响而感觉它在运动的现象。如在流动着的云中看月亮时,倒觉得好像是月亮在移动,而云反而似乎是静止的。自动运动就是当我们在黑暗的室内凝视一发光小点,会感到小光点在运动的现象。当我们凝视飞落的瀑布时,在视线离开瀑布移向另外方向,会看到静止物体似乎在向着瀑布落下相反的方向运动,这种现象叫运动残像。

1. 实际运动

物体运动或观察者本身运动产生的运动感觉并没有特别的区别。为了感知物体运动，运动的速度和运动的偏移量必须大于某一数值，但速度过大也不会有运动的感觉。各种情况下的界限值叫做速度门限、运动距离门限和速度极限。在速度变化或者二个运动物体有速度差的情况下，能感觉到运动的最小速度差叫做速度辨别门限。速度辨别门限与测试目标大小、背景情况及明暗条件有关。运动距离门限用视角来表示，其最小值约 8″～20″。关于速度门限，一般以一个小点的移动为目标来实验，当背景质地不同时角速度约为 1～2(′)/s，当背景质地相同时约为上述值的 10～20 倍。

中心视觉与边缘视觉相比较，一般来说中心视觉速度门限低，速度极限高。因此，仍然是中心视觉最好。感觉速度快慢的门限值同样受各种因素影响。例如，视线跟踪物体与视线固定两种情况后者感觉快；如果物体的形状在运动方向上较长和与运动方向相垂直的方向上较长的两物体以相同速度运动，会感觉前者速度快；大物体和小物体相比，当它们以相等的速度运动时，我们会感到大物体运动速度较慢。

2. 假象运动

一般把静止的光刺激的交替发生而产生运动的感觉叫假象运动。例如，把两个红灯泡一个点亮，一个关掉这样反复切换，会感到好像一个点亮的红灯泡在往复运动。假象运动最为简单的情况是二个点光源在适当的距离间隔和适当的时间间隔交替点亮和关掉而产生，其间隔和点亮的时间不同，感觉运动的情况也不同，把感觉运动最平稳的时间条件叫最佳时间。

科特(Korte)(1915 年)对于刺激出现的时间 T，距离间隔 S，刺激强度 I，时间间隔 P 的关系进行了研究。保持一恒定的运动感觉，当 I 增大时，可用增大 S 来补偿(第一法则)；P 增大时，可用 I 增大来补偿(第二法则)；P 增大时，可用 S 增大来补偿(第三法则)，也可用减小 T 来补偿(第四法则)。

假象运动如果加进图形因素，运动方向就会受到显著影响。如图 2-51(a)所示的图形，如果○和●的刺激图形交替地出现和消失，就会感到图形在以 AB 为轴作立体旋转运动(如同翻书一样的运动)。图 2-51(b)是图形影响的另一例子。当虚线图形和实线图形交替出现时会产生逆时针方向运动的感觉而绝不会出现相反的运动的感觉。

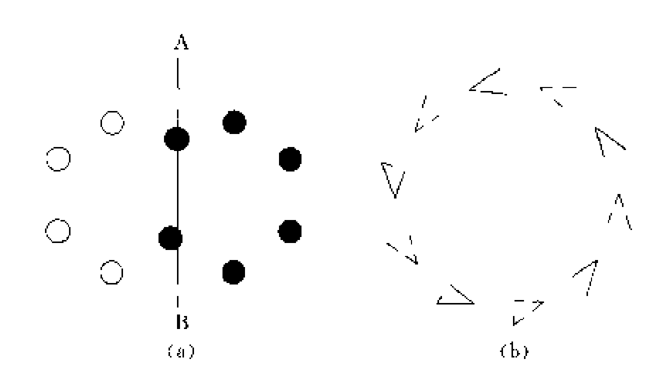


图 2-51　假象运动图形影响的实例

3. 残像

光刺激移去后而残留的感觉叫残像。残象形成直到消灭的经过是复杂的。白色光刺激的痕迹一般是明像和暗像交替出现，徐徐消灭，有色光刺激后是刺激光的颜色和它的补色相近的色交替出现徐徐消灭。残像主要是由视觉细胞的光化学反应引起的。

2.8 形状感觉与错视

视觉系统所感觉到的物体的形状并不是简单的投影到视网膜上的原封不动的形状。对形状的感觉受到物体自身形状及周围背景的影响。这类影响是多种多样的，有神经系统引起的错视现象也有心理因素的作用。

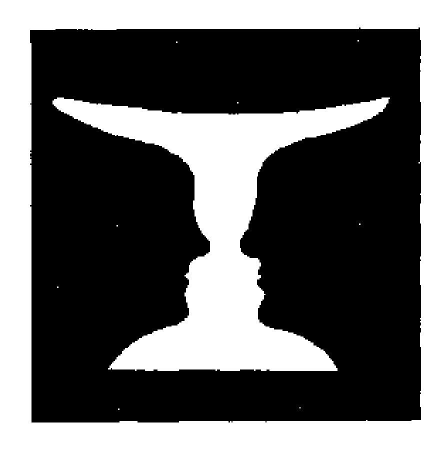


图 2-52 图和背景反转的图形

在研究心理学的作用方面格斯特尔特(Gestalt)学派曾作出了较大的贡献。比如沃梯默(Wertheimer)提出，当出示几个图形时，互相接近的人之间对图形的感觉比较容易取得一致的看法，这里就包括有心理上的诱导作用，他把这种现象叫做群化法则。另一个关系到心理因素的重要问题是图形和背景的关系问题。例如，看到图 2-52 的图形时，首先感觉到的是图形还是背景呢？通过研究发现，对图形和背景的感觉与观察者的经验、态度、明暗差别以及面积的比例等各种因素都有关系。

错视是视觉对图形感觉的一个重要现象。图 2-53 是几个著名的几何学的错视图形的例子。图(a)本是两条相等的线段，由于两端加了不同方向的图形使我们感觉下边的一条线段较长；图(b)使我们看到斜线是错位的；图(c)中原本是二条平行的直线，可给我们的感觉却是二根弯曲的线；图(d)本来是互相平行的三条线，可我们看到的却是不平行的了；图(e)中左边和右边两图中央的圆是相同的，但我们都觉得右边的要大。所有这些均是由错视造成的。

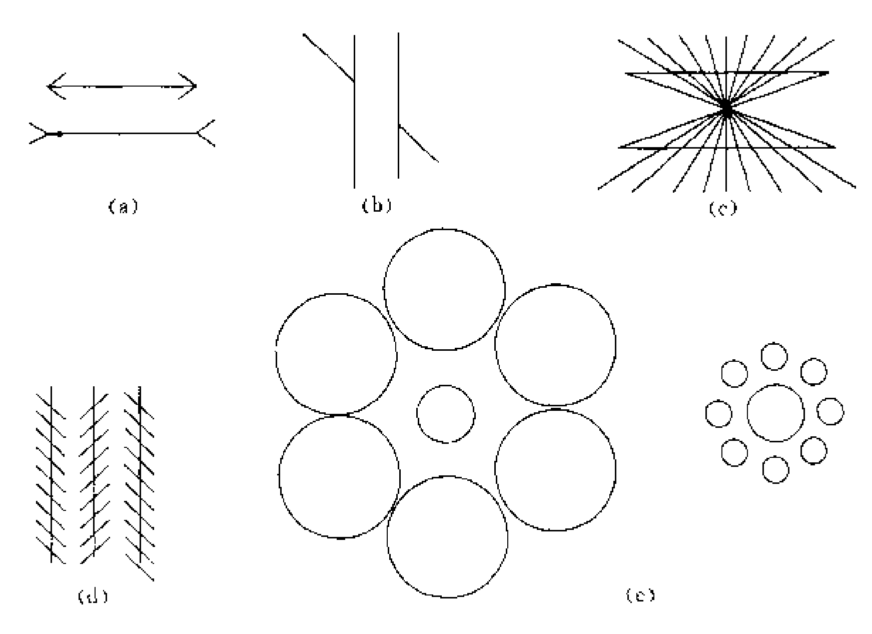


图 2-53 几种错视图形

视觉对大小形状的感觉也受时间因素的影响。例如，当我们长时间凝视一条弯曲的线之后马上看一条直线，就会感到这条直线向着原来所看的曲线相反的方向弯曲。这种现象叫做吉布森(Gibson)效应。另外，如图 2-54 所示的两个同心圆，外边的圆的直径固定，当内部小圆

的直径作大小变化时，我们看到的情况却好像大圆也在变化。通常这个现象叫图形残效(figural left effect)。

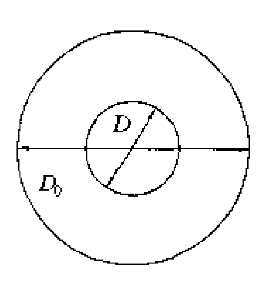
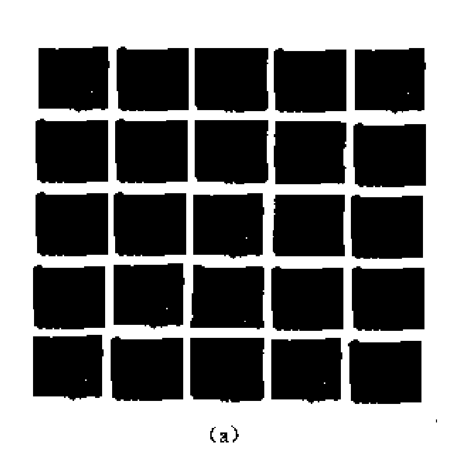
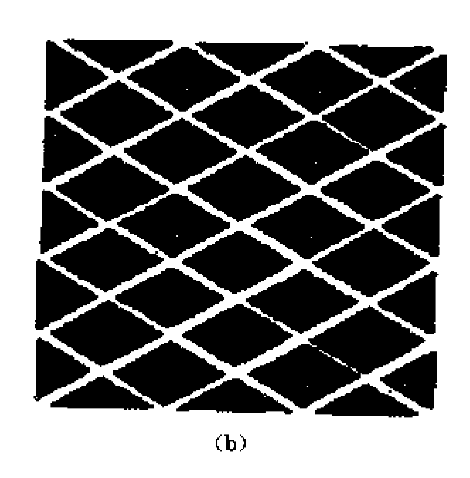


图 2-54　同心圆过大过小的错视

当视网膜受到刺激时，并不是只有与刺激部位相对应的神经系统产生反应，而是对其周围也有影响。这种影响可以看成是由某种场引起的，所以把它叫做诱导场。利用诱导场的概念可以解释一些错视现象。图 2-55 所示的图形，(a)中的白色线条的交点处给我们的感觉是灰色的而不是白色的。图(b)中菱形的中央特别黑。这些现象都可以用诱导场的概念来解释。



(a)



(b)

图 2-55　诱导场引起的视觉现象

在这一章中，我们对图像和视觉的特性进行了许多讨论。但这只是一些初步认识，应该看到，就图像本身的客观性质而言，至今尚未找到一个更加贴切的数学模型来表达图像的内在实质，同时对于视觉器官以及人的生理和心理特性的研究也远未穷尽。在图像处理这一领域中会涉及许多边缘学科知识，若在这些边缘学科中有新的突破，必然会给图像处理这一学科带来方向性的影响。所以，在深入研究各种处理方法的同时，要对图像信号的统计特性及视觉特性这一带有根本性的理论问题给予充分的注意

思　考　题

1. 光学中的主要计量单位有哪些？它们的含义是什么？
2. 图像的统计特性包括哪些内容？
3. 图像信号的自相关函数 $\rho(\xi,\eta)$ 与 $|C(u,v)|^2$ 是什么关系？
4. 图像的功率谱密度 $\phi(u,v)$ 与自相关函数是什么关系？
5. 何为图像信息的熵？离散图像信息与连续图像信息的熵有何区别？其表达式是什么？
6. 摄像器件的性能从哪几个方面考虑？
7. 什么叫光觉？什么叫色觉？
8. 什么是三基色？相加混色与相减混色的基色是否相同？
9. 格拉斯曼定律包含哪些内容？
10. 标准白光有哪几种？其色度坐标是多少？

11. 何为第三色盲？它对图像处理有何意义？
12. 何为视力？视力与哪些因素有关？
13. 何为马赫现象？它指什么而言？
14. 何为光觉门限？
15. 错视现象对图像处理有何意义？

第 3 章　图像处理中的正交变换

数字图像处理的方法主要分为两大类：一类是空间域处理法(或称空域法)，一类是频域法(或称变换域法)。在频域法处理中最为关键的预处理便是变换处理。这种变换一般是线性变换，其基本线性运算式是严格可逆的，并且满足一定的正交条件，因此，也将其称作酉变换。目前，在图像处理技术中正交变换被广泛地运用于图像特征提取、图像增强、图像复原、图像识别以及图像编码等处理中。本章将对几种主要的正交变换进行较详细地讨论。

3.1　傅里叶变换

傅里叶变换是大家所熟知的正交变换。在一维信号处理中得到了广泛应用。把这种处理方法推广到图像处理中是很自然的事。本节将对傅里叶变换的基本概念及算法作一些讨论。

3.1.1　傅里叶变换的定义及基本概念

傅里叶变换在数学中的定义是严格的。设 $f(x)$ 为 x 的函数，如果 $f(x)$ 满足下面的狄里赫莱条件：

(1) 具有有限个间断点；

(2) 具有有限个极值点；

(3) 绝对可积。

则有下列二式成立：

$$F(u) = \int_{-\infty}^{+\infty} f(x)e^{-j2\pi ux}dx \tag{3-1}$$

$$f(x) = \int_{-\infty}^{+\infty} F(u)e^{j2\pi ux}du \tag{3-2}$$

式中 x 为时域变量，u 为频率变量。如果令 $\omega=2\pi u$，则有：

$$F(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-j\omega x}dx \tag{3-3}$$

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{j\omega x}d\omega \tag{3-4}$$

通常把以上公式称为傅里叶变换对。

函数 $f(x)$ 的傅里叶变换一般是一个复量，它可以由式(3-5)表示：

$$F(\omega) = R(\omega) + jI(\omega) \tag{3-5}$$

或写成指数形式：

$$F(\omega) = |F(\omega)|e^{j\phi(\omega)} \tag{3-6}$$

$$|F(\omega)| = \sqrt{R^2(\omega) + I^2(\omega)} \tag{3-7}$$

$$\phi(\omega) = \arctan\frac{I(\omega)}{R(\omega)} \tag{3-8}$$

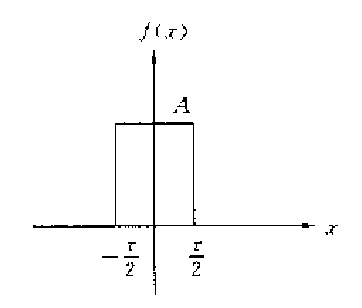


图 3-1　函数 $f(x)$的波形

$|F(\omega)|$称做 $f(x)$的傅里叶谱，而 $\phi(\omega)$称为相位谱。

傅里叶变换广泛用于频谱分析。

例：求图 3-1 所示波形 $f(x)$的频谱。

$$f(x)=\begin{cases} A & -\frac{\tau}{2}\leqslant x\leqslant\frac{\tau}{2} \\ 0 & x>\frac{\tau}{2} \\ 0 & x<\frac{\tau}{2} \end{cases}$$

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{+\infty} f(x)\mathrm{e}^{-\mathrm{j}\omega x}\mathrm{d}x \\ &= \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} A\mathrm{e}^{-\mathrm{j}\omega x}\mathrm{d}x \\ &= \frac{A}{\mathrm{j}\omega}(\mathrm{e}^{\mathrm{j}\omega\frac{\tau}{2}} - \mathrm{e}^{-\mathrm{j}\omega\frac{\tau}{2}}) \\ &= \frac{2A}{\omega}\sin\frac{\omega\tau}{2} \end{aligned}$$

则

$$\begin{aligned} |F(\omega)| &= \frac{2A}{\omega}\left|\sin\frac{\omega\tau}{2}\right| \\ &= A\tau\left|\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}\right| \end{aligned}$$

$$\phi(\omega)=\begin{cases} 0 & \frac{4n\pi}{\tau}<\omega<\frac{2(2n+1)\pi}{\tau},\ n=0,1,2,\cdots \\ \pi & \frac{2(2n+1)\pi}{\tau}<\omega<\frac{4(n+1)\pi}{\tau},\ n=0,1,2,\cdots \end{cases}$$

$f(x)$的幅度谱及相位谱如图 3-2 所示。

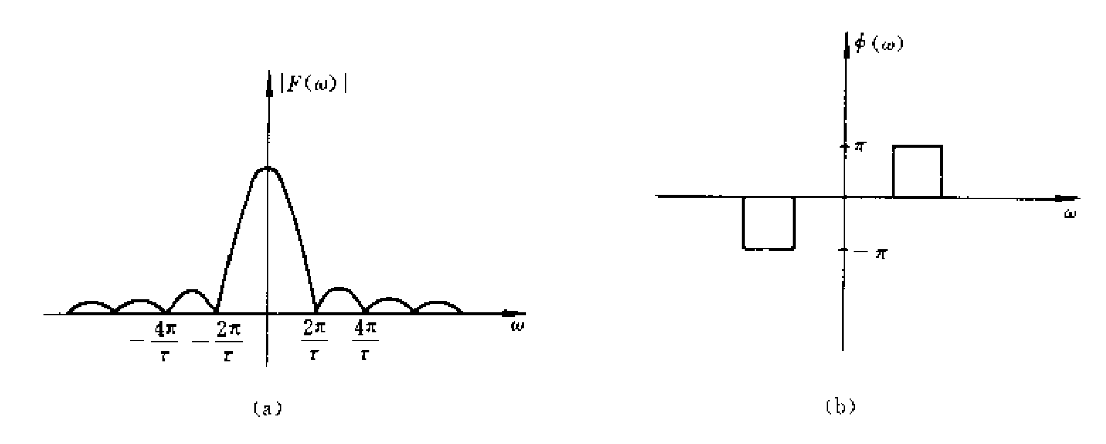


图 3-2　$f(x)$的幅度谱及相位谱

例　求周期函数的傅里叶谱：

一个周期为 T 的信号 $f(x)$可用傅里叶级数来表示，即

$$f(x)=\sum_{n=-\infty}^{+\infty}F(n)e^{-jn\omega_0 x}$$

$$F(n)=\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}f(x)e^{-jn\omega_0 x}dx$$

式中

$$\omega_0=\frac{2\pi}{T}$$

因此，傅里叶变换可写成下式：

$$\begin{aligned}
F(\omega)&=\mathscr{F}[f(x)]\\
&=\mathscr{F}\left[\sum_{n=-\infty}^{+\infty}F(n)e^{-jn\omega_0 x}\right]\\
&=\sum_{n=-\infty}^{+\infty}F(n)\mathscr{F}[e^{-jn\omega_0 x}]\\
&=\sum_{n=-\infty}^{+\infty}F(n)\int_{-\infty}^{+\infty}e^{jn\omega_0 x}\cdot e^{-j\omega x}dx\\
&=\sum_{n=-\infty}^{+\infty}F(n)\int_{-\infty}^{+\infty}e^{-j(\omega-n\omega_0)x}dx\\
&=2\pi\sum_{n=-\infty}^{+\infty}F(n)\delta(\omega-n\omega_0)
\end{aligned}$$

式中 $\delta(\omega-n\omega_0)$ 是冲激序列，其幅度谱如图 3-3 所示。

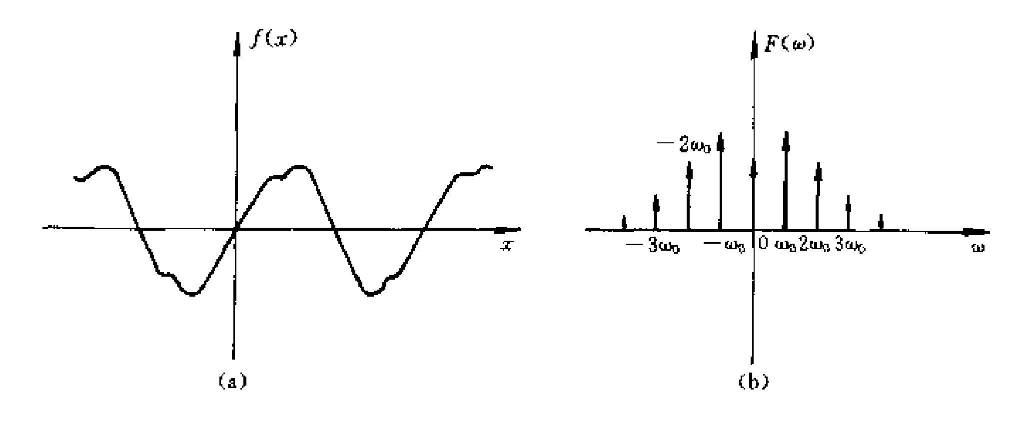


图 3-3　周期函数的傅里叶谱

由上面的例子可以建立起下面几个概念：

1)只要满足狄里赫莱条件，连续函数就可以进行傅里叶变换，实际上这个条件在工程运用中总是可以满足的。

2)连续非周期函数的傅里叶谱是连续的非周期函数，连续的周期函数的傅里叶谱是离散的非周期函数。

傅里叶变换可推广到二维函数。如果二维函数 $f(x,y)$ 满足狄里赫莱条件，那么将有下面二维傅里叶变换对存在：

$$F(u,v)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(x,y)e^{-j2\pi(ux+vy)}dxdy \tag{3-9}$$

$$f(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}F(u,v)\mathrm{e}^{\mathrm{j}2\pi(ux+vy)}\mathrm{d}u\mathrm{d}v \tag{3-10}$$

与一维傅里叶变换类似，二维傅里叶变换的幅度谱和相位谱如下式：

$$|F(u,v)|=\sqrt{R^2(u,v)+I^2(u,v)} \tag{3-11}$$

$$\phi(u,v)=\arctan\frac{I(u,v)}{R(u,v)} \tag{3-12}$$

$$E(u,v)=R^2(u,v)+I^2(u,v) \tag{3-13}$$

式中：$F(u,v)$是幅度谱，$\phi(u,v)$是相位谱，$E(u,v)$是能量谱。

例　求图 3-4 所示函数的傅里叶谱。

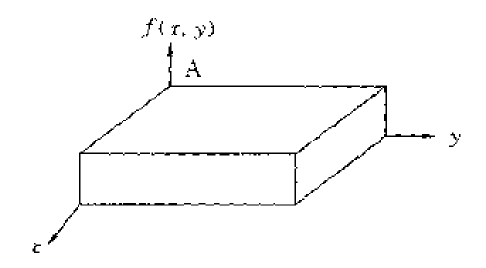


图 3-4　函数 $f(x,y)$

$$f(x,y)=\begin{cases}A & 0\leqslant x\leqslant X,\ 0\leqslant y\leqslant Y\\ 0 & x>X,\ x<0;\ y>Y,\ y<0\end{cases}$$

$$\begin{aligned}F(u,v)&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y\\&=\int_0^X\int_0^Y A\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y\\&=A\int_0^X\mathrm{e}^{-\mathrm{j}2\pi ux}\mathrm{d}x\int_0^Y\mathrm{e}^{-\mathrm{j}2\pi vy}\mathrm{d}y\\&=A\left[-\frac{\mathrm{e}^{-\mathrm{j}2\pi ux}}{\mathrm{j}2\pi u}\right]_0^X\left[-\frac{\mathrm{e}^{-\mathrm{j}2\pi vy}}{\mathrm{j}2\pi v}\right]_0^Y\\&=\left(-\frac{A}{\mathrm{j}2\pi u}\right)\left[\mathrm{e}^{-\mathrm{j}2\pi uX}-1\right]\left(-\frac{1}{\mathrm{j}2\pi v}\right)\left[\mathrm{e}^{-\mathrm{j}2\pi vY}-1\right]\\&=AXY\left[\frac{\sin(\pi uX)\mathrm{e}^{-\mathrm{j}\pi uX}}{\pi uX}\right]\left[\frac{\sin(\pi vY)\mathrm{e}^{-\mathrm{j}\pi vY}}{\pi vY}\right]\end{aligned}$$

其傅里叶谱由下式表示：

$$|F(u,v)|=AXY\left|\frac{\sin(\pi uX)}{\pi uX}\right|\left|\frac{\sin(\pi vY)}{\pi vY}\right|$$

3.1.2　傅里叶变换的性质

傅里叶变换有许多重要性质，这些性质为实际运算处理提供了极大的便利。这里仅就二维傅里叶变换为例列出其主要的几个性质。

（1）可分性

$$\begin{aligned}F(u,v)&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi(ux+vy)}\mathrm{d}x\mathrm{d}y\\&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi ux}\mathrm{e}^{-\mathrm{j}2\pi vy}\mathrm{d}x\mathrm{d}y\end{aligned}$$

$$= \int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi ux}dx\right]e^{-j2\pi vy}dy \tag{3-14}$$

$$= \int_{-\infty}^{+\infty}\{\mathscr{F}_x[f(x,y)]\}e^{-j2\pi vy}dy$$

$$= \mathscr{F}_y\{\mathscr{F}_x[f(x,y)]\}$$

这个性质说明一个二维傅里叶变换可用二次一维傅里叶变换来实现。

(2) 线性

傅里叶变换是线性算子，即

$$\mathscr{F}[a_1f_1(x,y) + a_2f_2(x,y)]$$
$$= a_1\mathscr{F}[f_1(x,y)] + a_2\mathscr{F}[f_2(x,y)] \tag{3-15}$$

(3) 共轭对称性

如果 $F(u,v)$是 $f(x,y)$的傅里叶变换，$F^*(-u,-v)$是 $f(-x,-y)$傅里叶变换的共轭函数，那么，

$$F(u,v) = F^*(-u,-v) \tag{3-16}$$

(4) 旋转性

如果空间域函数旋转的角度为 θ_0，那么在变换域中此函数的傅里叶变换也旋转同样的角度，即

$$f(r,\theta+\theta_0) \Leftrightarrow F(k,\phi+\theta_0) \tag{3-17}$$

在式(3-17)中引入极坐标表示。其中：$x=r\cos\theta$，$y=r\sin\theta$，$u=k\cos\phi$，$v=k\sin\phi$。所以 $f(x,y)$和 $F(u,v)$分别用 $f(r,\theta)$和 $F(k,\phi)$来表示。式中的⇔为对应关系符号。反之，如果 $F(u,v)$旋转某一角度，则 $f(x,y)$在空间域也旋转同样的角度。这条性质只要以极坐标代以 x,y,u,v，则可以得到证明。

(5) 比例变换特性

如果 $F(u,v)$是 $f(x,y)$的傅里叶变换，a 和 b 分别为两个标量，那么

$$af(x,y) \Leftrightarrow aF(u,v) \tag{3-18}$$

$$f(ax,by) \Leftrightarrow \frac{1}{|ab|}F\left(\frac{u}{a},\frac{v}{b}\right) \tag{3-19}$$

(6) 帕斯维尔(Parseval)定理

这个性质也可称为能量保持定理。如果 $F(u,v)$是 $f(x,y)$的傅里叶变换，那么有下式成立

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|f(x,y)|^2dxdy = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|F(u,v)|^2dudv \tag{3-20}$$

这个性质说明变换前后并不损失能量。

(7) 相关定理

如果 $f(x)$，$g(x)$为两个一维时域函数，$f(x,y)$和 $g(x,y)$为两个二维空域函数，那么，定义下两式为相关运算：

$$f(x)\circ g(x) = \int_{-\infty}^{+\infty} f(\alpha)g(x+\alpha)d\alpha \tag{3-21}$$

$$f(x,y)\circ g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)g(x+\alpha,y+\beta)d\alpha d\beta \tag{3-22}$$

式中，$\circ$ 符号表示相关运算。由以上定义可引出傅里叶变换的一个重要性质，这就是相关定理，即

$$f(x,y) \circ g(x,y) \Leftrightarrow F(u,v) \cdot G^*(u,v) \tag{3-23}$$

$$f(x,y) \cdot g^*(x,y) \Leftrightarrow F(u,v) \circ G(u,v) \tag{3-24}$$

式中 $F(u,v)$是 $f(x,y)$的傅里叶变换，$G(u,v)$是 $g(x,y)$的傅里叶变换，$G^*(u,v)$是 $G(u,v)$的共轭，$g^*(x,y)$是 $g(x,y)$的共轭。

(8) 卷积定理

如果 $f(x)$和 $g(x)$是一维时域函数，$f(x,y)$和 $g(x,y)$是二维空域函数，那么，定义以下两式为卷积运算，即

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(\alpha) g(x-\alpha) \mathrm{d}\alpha \tag{3-25}$$

$$f(x,y) * g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta) g(x-\alpha, y-\beta) \mathrm{d}\alpha \mathrm{d}\beta \tag{3-26}$$

式中 * 符号表示卷积关系。由此，可得到傅里叶变换的卷积定理如下：

$$f(x,y) * g(x,y) \Leftrightarrow F(u,v) \cdot G(u,v) \tag{3-27}$$

$$f(x,y) \cdot g(x,y) \Leftrightarrow F(u,v) * G(u,v) \tag{3-28}$$

式中 $F(u,v)$和 $G(u,v)$分别是 $f(x,y)$和 $g(x,y)$的傅里叶变换。

2.1.3 离散傅里叶变换

连续函数的傅里叶变换是波形分析的有力工具，这在理论分析中无疑具有很大的价值。离散傅里叶变换使得数学方法与计算机技术建立了联系，为傅里叶变换这样一个数学工具在实用中开辟了一条宽阔的道路。因此，它不仅具有理论价值，而且在某种意义上说它也有了更要的实用价值。

1. 离散傅里叶变换的定义

如果 $x(n)$为一数字序列，则其离散傅里叶正变换定义由下式来表示：

$$X(m) = \sum_{n=0}^{N-1} x(n) \mathrm{e}^{-\mathrm{j}\frac{2\pi mn}{N}} \tag{3-29}$$

傅里叶反变换定义由下式来表示：

$$x(n) = \frac{1}{N}\sum_{m=0}^{N-1} X(m) \mathrm{e}^{\mathrm{j}\frac{2\pi mn}{N}} \tag{3-30}$$

由式(3-29)和式(3-30)可见，离散傅里叶变换是直接处理离散时间信号的傅里叶变换。如果要对一个连续信号进行计算机数字处理，那么就必须经过离散化处理。这样，对连续信号进行的傅里叶变换的积分过程就会自然地蜕变为求和过程。关于这一点，下面先用示意图图 3-5 建立一个直观概念。由图(a)可见，时域信号是非周期性的连续信号，其傅里叶谱就是连续的非周期性的波形。由图(b)可见，时域信号是周期性的连续信号，其傅里叶谱就是非周期性的离散谱。由图(c)可见，将时域信号通过取样作离散化处理，其傅里叶谱就是周期性的连续谱。从图(d)可见，将时域信号作离散化处理并延拓为周期性信号，其傅里叶变换就是离散的周期性的谱了，以此可以看出离散傅里叶变换的概念。

上述概念也可以用数学方式表述如下：

如果 $x(t)$是连续的非周期函数，其频谱 $X(f)$就是连续的非周期谱。由图(b)可见，对周期函数来说，其傅里叶变换可写成式(3-31)和式(3-32)：

$$X(m) = \frac{1}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} x(t) \mathrm{e}^{-\mathrm{j}2\pi mt} \mathrm{d}t \tag{3-31}$$

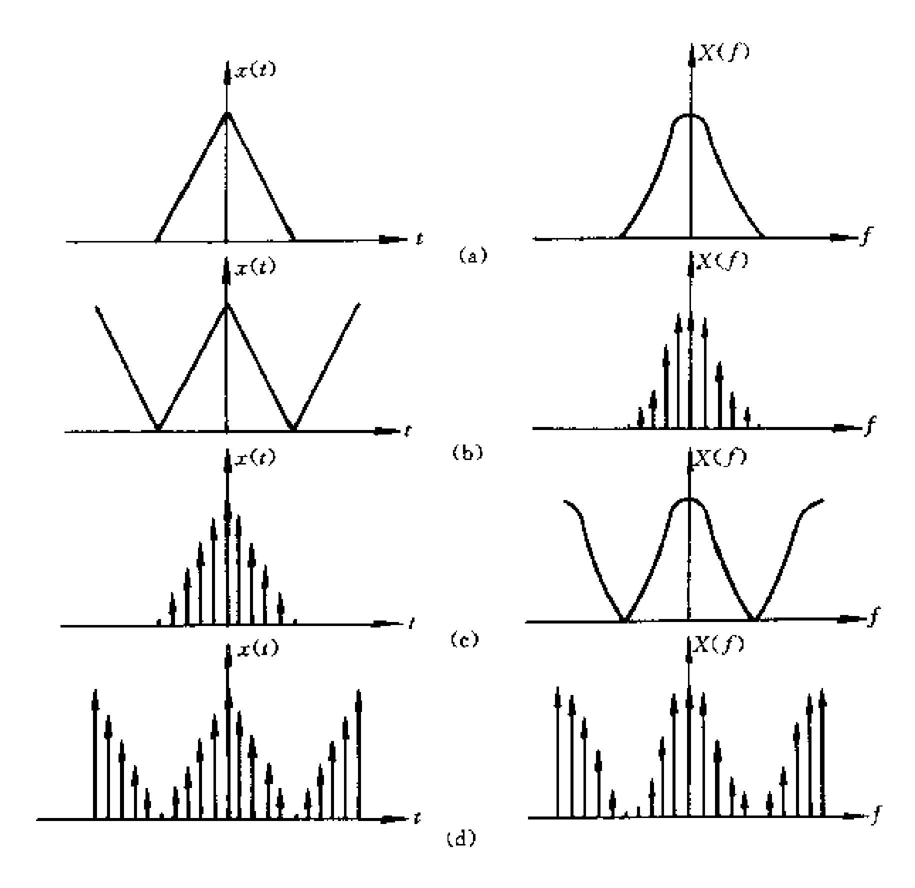


图 3-5　离散傅里叶变换的物理概念示意图

$$x(t)=\sum_{m=-\infty}^{+\infty} X(m)\mathrm{e}^{\mathrm{j}2\pi m t} \tag{3-32}$$

如果 $x(t)$ 是离散的函数，那么，其频谱就是一个周期的连续谱，可写成式(3-33)和式(3-34)：

$$X(f)=\sum_{n=-\infty}^{+\infty} x(n)\mathrm{e}^{-\mathrm{j}2\pi n f} \tag{3-33}$$

$$x(n)=\frac{1}{f}\int_{-\frac{f}{2}}^{+\frac{f}{2}} X(f)\mathrm{e}^{\mathrm{j}2\pi n f}\mathrm{d}f \tag{3-34}$$

综合上述二种情况，如果 $x(n)$ 既是离散的，又是周期的函数，那么，它的傅里叶谱则必然是离散的、周期的谱，即

$$X(m)=\sum_{n=-\infty}^{+\infty} x(n)\mathrm{e}^{-\mathrm{j}2\pi m n} \tag{3-35}$$

$$x(n)=\sum_{m=-\infty}^{+\infty} X(m)\mathrm{e}^{\mathrm{j}2\pi m n} \tag{3-36}$$

由上面的讨论可以比较容易地建立起离散傅里叶变换的物理概念。正因为函数从连续函数变为离散函数，所以，也就使傅里叶变换的积分运算变为求和运算。

2. 离散傅里叶变换的性质

离散傅里叶变换有如下一些性质。

(1) 线性

如果时间序列 $x(n)$ 与 $y(n)$ 各有其傅里叶变换 $X(m)$ 和 $Y(m)$，则

$$ax(n)+by(n)\Leftrightarrow aX(m)+bY(m) \tag{3-37}$$

(2) 对称性

如果 $$x(n)\Leftrightarrow X(m)$$

则 $$\frac{1}{N}X(n)\Leftrightarrow x(-m) \tag{3-38}$$

证明　因为 $$x(n)=\frac{1}{N}\sum_{m=0}^{N-1}X(m)\cdot W^{-mn}$$

这里令 $W=e^{-j\frac{2\pi}{N}}$，更换 m 与 n，则

$$x(m)=\frac{1}{N}\sum_{n=0}^{N-1}X(n)\cdot W^{-mn}$$

$$x(-m)=\sum_{n=0}^{N-1}\frac{1}{N}X(n)\cdot W^{mn}$$

由此可知，$x(-m)$与$\frac{1}{N}X(n)$是一对傅里叶变换关系，所以得：

$$\frac{1}{N}X(n)\Leftrightarrow x(-m) \tag{3-39}$$

(3) 时间移位

如果序列向右(或向左)移动 k 位，则：

$$x(n-k)\Leftrightarrow X(m)\cdot W^{km}$$

证明　因为

$$\begin{aligned}x(n-k)&=\frac{1}{N}\sum_{m=0}^{N-1}X(m)\cdot W^{-(n-k)m}\\&=\frac{1}{N}\sum_{m=0}^{N-1}X(m)\cdot W^{-nm}\cdot W^{km}\\&=\frac{1}{N}\sum_{m=0}^{N-1}[X(m)\cdot W^{km}]\cdot W^{-mn}\end{aligned}$$

则 $$x(n-k)\Leftrightarrow X(m)\cdot W^{km}$$

(4) 频率移位

如果 $$x(n)\Leftrightarrow X(m)$$

则 $$x(n)\cdot W^{-kn}\Leftrightarrow X(m-k) \tag{3-40}$$

证明　因为 $$X(m)=\sum_{n=0}^{N-1}x(n)\cdot W^{mn}$$

$$\begin{aligned}x(m-k)&=\sum_{n=0}^{N-1}x(n)\cdot W^{(m-k)n}\\&=\sum_{n=0}^{N-1}x(n)\cdot W^{mn}\cdot W^{-kn}\\&=\sum_{n=0}^{N-1}[x(n)\cdot W^{-kn}]\cdot W^{mn}\end{aligned}$$

则 $$x(n)\cdot W^{-kn}\Leftrightarrow X(m-k)$$

(5) 周期性

如果 $$x(n)\Leftrightarrow X(m)$$

则 $$x(n \pm rN) = x(n) \tag{3-41}$$

证明 $$x(n \pm rN) = \frac{1}{N}\sum_{m=0}^{N-1} X(m) \cdot W^{-(n \pm rN)m}$$

$$= \frac{1}{N}\sum_{m=0}^{N-1} X(m) \cdot W^{\mp mrN} \cdot W^{-mn}$$

由于 $$W = e^{-j\frac{2\pi}{N}}$$

因此 $$W^{\pm mrN} = e^{-j\frac{2\pi}{N}mrN} = e^{-j2\pi mr} = 1 \quad (r\text{ 是正整数})$$

代入前式中有 $$X(m) \cdot W^{-mrN} = X(m)$$

则 $$x(n \pm rN) = x(n)$$

(6) 偶函数

如果 $$x_e(n) \Leftrightarrow x_e(-n)$$

则 $$X_e(m) = \sum_{n=0}^{N-1} x_e(n)\cos\left(\frac{2\pi mn}{N}\right) \tag{3-42}$$

证明 因为 $$X(m) = \sum_{n=0}^{N-1} x_e(n)e^{-j\frac{2\pi}{N}mn}$$

$$= \sum_{n=0}^{N-1} x_e(n)\left[\cos\left(\frac{2\pi}{N}mn\right) - j\sin\left(\frac{2\pi}{N}mn\right)\right]$$

$$= \sum_{n=0}^{N-1} x_e(n)\left[\cos\left(\frac{2\pi}{N}mn\right) - j\sum_{n=0}^{N-1} x_e(n)\sin\left(\frac{2\pi}{N}mn\right)\right]$$

由于 $x_e(n) = x_e(-n)$ 是一偶函数，$\cos\left(\frac{2\pi}{N}mn\right)$ 是偶函数。$\sin\left(\frac{2\pi}{N}mn\right)$ 是奇函数，所以上式中的虚部累加和是 0，则

$$X_e(m) = \sum_{n=0}^{N-1} x_e(n) \cdot \cos\left(\frac{2\pi}{N}mn\right)$$

(7) 奇函数

如果 $$x_o(n) = -x_o(-n)$$

则 $$X_o(m) = -j\sum_{n=0}^{N-1} x_o(n) \cdot \sin\left(\frac{2\pi}{N}mn\right) \tag{3-43}$$

证明

$$X_o(m) = \sum_{n=0}^{N-1} x_o(n)e^{-j\frac{2\pi}{N}mn}$$

$$= \sum_{n=0}^{N-1} x_o(n)\left[\cos\left(\frac{2\pi}{N}mn\right) - j\sin\left(\frac{2\pi}{N}mn\right)\right]$$

$$= \sum_{n=0}^{N-1} x_o(n) \cdot \cos\left(\frac{2\pi}{N}mn\right) - j\sum_{n=0}^{N-1} x_o(n) \cdot \sin\left(\frac{2\pi}{N}mn\right)$$

由于 $x_o(n)$ 是奇函数，而 $\cos\left(\frac{2\pi}{N}mn\right)$ 是偶函数，因此，实部之和为 0，则

$$X_o(m) = -j\sum_{n=0}^{N-1} x_o(n) \cdot \sin\left(\frac{2\pi}{N}mn\right)$$

(8) 卷积定理

如果 $$x(n) \Leftrightarrow X(m), \quad y(n) \Leftrightarrow Y(m)$$

则 $$x(n) * y(n) \Leftrightarrow X(m) \cdot Y(m) \tag{3-44}$$

证明 由卷积定义可知，

$$x(n) * y(n) = \sum_{h=0}^{N-1} x(h) y(n-h)$$

又设 $x(n) * y(n)$的离散傅里叶变换为 C，

则

$$C = \sum_{n=0}^{N-1} [x(n) * y(n)] \cdot W^{mn}$$

$$= \sum_{n=0}^{N-1} \Big[\sum_{h=0}^{N-1} x(h) y(n-h) \Big] \cdot W^{mn}$$

$$= \sum_{n=0}^{N-1} \sum_{h=0}^{N-1} x(h) y(n-h) \cdot W^{mn}$$

$$= \sum_{h=0}^{N-1} x(h) \Big[\sum_{n=0}^{N-1} y(n-h) \cdot W^{mn} \Big]$$

由移位定理可知：

$$\sum_{n=0}^{N-1} y(n-h) \cdot W^{mn} = \Big[\sum_{n=0}^{N-1} y(n) \cdot W^{mn} \Big] \cdot W^{mh}$$

代入上式有：

$$C = \Big[\sum_{h=0}^{N-1} x(h) \cdot W^{mh} \Big] \Big[\sum_{n=0}^{N-1} y(n) \cdot W^{mn} \Big]$$

$$= X(m) \cdot Y(m)$$

因此 $$x(n) * y(n) \Leftrightarrow X(m) \cdot Y(m)$$

反之 $$x(n) \cdot y(n) \Leftrightarrow X(m) * Y(m)$$

也成立。

(9) 相关定理

如果 $$x(n) \Leftrightarrow X(m)$$

$$y(n) \Leftrightarrow Y(m)$$

则 $$x(n) \circ y(n) \Leftrightarrow X^*(m) \cdot Y(m) \tag{3-45}$$

证明 由相关定义有：

$$x(n) \circ y(n) = \sum_{h=0}^{N-1} x(h) y(n+h)$$

设相关的傅里叶变换为 C'，则

$$C' = \sum_{n=0}^{N-1} [x(n) \circ y(n)] \cdot W^{mn}$$

$$= \sum_{n=0}^{N-1} \Big[\sum_{h=0}^{N-1} x(h) y(n+h) \Big] \cdot W^{mn}$$

$$= \sum_{h=0}^{N-1} x(h) \Big[\sum_{n=0}^{N-1} y(n+h) \cdot W^{mn} \Big]$$

由移位性质，

$$\sum_{n=0}^{N-1} y(n+h) W^{mn} = \sum_{n=0}^{N-1} y(n) W^{mn} W^{-mh}$$

代入上式有

$$C' = \sum_{h=0}^{N-1} x(h) \cdot W^{-mh} \cdot \sum_{n=0}^{N-1} y(n) \cdot W^{mn}$$

$$= X^*(m) \cdot Y(m)$$

其中，
$$X(m) = \sum_{h=0}^{N-1} x(h) \cdot W^{mh}$$

$$X^* = \sum_{h=0}^{N-1} x(h) \cdot W^{-mh}$$

则
$$x(n) \circ y(n) \Leftrightarrow X^*(m) \cdot Y(m)$$

(10) 帕斯维尔定理

如果
$$x(n) \Leftrightarrow X(m)$$

则
$$\sum_{n=0}^{N-1} x^2(n) = \frac{1}{N} \sum_{m=0}^{N-1} |X(m)|^2 \tag{3-46}$$

证明 由相关定理，如果 $x(n)=y(n)$，则 $C'=|X(m)|^2$，求 C' 的离散傅里叶反变换，

则
$$\frac{1}{N} \sum_{m=0}^{N-1} |X(m)|^2 \cdot W^{-nm} = \sum_{h=0}^{N-1} x(h) x(n+h)$$

令 $n=0$，

则
$$\sum_{h=0}^{N-1} x^2(h) = \frac{1}{N} \sum_{m=0}^{N-1} |X(m)|^2$$

3.1.4 快速傅里叶变换(FFT)

随着计算机技术和数字电路的迅速发展，在信号处理中使用计算机和数字电路的趋势愈加明显。离散傅里叶变换已成为数字信号处理的重要工具。然而，它的计算量较大，运算时间长，在某种程度上却限制了它的使用范围。快速算法大大提高了运算速度，在某些应用场合已可能作到实时处理，并且开始应用于控制系统。快速傅里叶变换并不是一种新的变换，它是离散傅里叶变换的一种算法。这种方法是在分析离散傅里叶变换中的多余运算的基础上，进而消除这些重复工作的思想指导下得到的，所以在运算中大大节省了工作量，达到了快速运算的目的。下面我们从基本定义入手，讨论其原理。

对于一个有限长序列 $\{x(n)\}(0 \leqslant n \leqslant N-1)$，它的傅里叶变换由下式表示：

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi mn}{N}} \quad m = 0,1,2,\cdots,N-1$$

令
$$W = e^{-j\frac{2\pi}{N}} \quad W^{-1} = e^{j\frac{2\pi}{N}} \tag{3-47}$$

因此，傅里叶变换对可写成下式：

$$X(m) = \sum_{n=0}^{N-1} x(n) W^{mn} \tag{3-48}$$

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) W^{-mn} \tag{3-49}$$

将正变换式(3-48)展开可得到如下算式：

$$\begin{aligned} X(0) &= x(0)W^{00} + x(1)W^{01} + \cdots + x(N-1)W^{0(N-1)} \\ X(1) &= x(0)W^{10} + x(1)W^{11} + \cdots + x(N-1)W^{1(N-1)} \\ X(2) &= x(0)W^{20} + x(1)W^{21} + \cdots + x(N-1)W^{2(N-1)} \\ &\cdots \end{aligned} \tag{3-50}$$

$$X(N-1)=x(0)W^{(N-1)0}+x(1)W^{(N-1)1}+\cdots+x(N-1)W^{(N-1)(N-1)}$$

上面的方程式(3-50)可以用矩阵表示：

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \cdots \\ X(N-1) \end{bmatrix}=\begin{bmatrix} W^{00} & W^{01} & \cdots & W^{0(N-1)} \\ W^{10} & W^{11} & \cdots & W^{1(N-1)} \\ W^{20} & W^{21} & \cdots & W^{2(N-1)} \\ \cdots & \cdots & \cdots & \cdots \\ W^{(N-1)0} & W^{(N-1)1} & \cdots & W^{(N-1)(N-1)} \end{bmatrix}\cdot\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \cdots \\ x(N-1) \end{bmatrix} \tag{3-51}$$

从上面的运算显然可以看出，要得到每一个频率分量，需进行 N 次乘法和 $N-1$ 次加法运算。要完成整个变换需要 N^2 次乘法和 $N(N-1)$次加法运算。当序列较长时，必然要花费大量的时间。

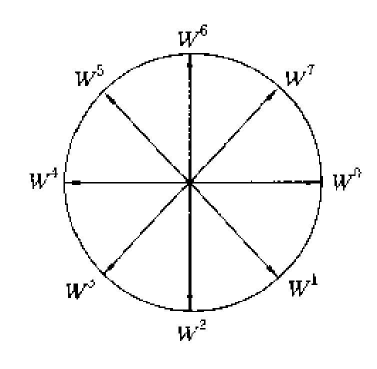


图 3-6 $N=8$ 时 W^{mn}的周期性和对称性

观察上述系数矩阵，发现 W^{mn}是以 N 为周期的，即

$$W^{(m+LN)(n+hN)}=W^{mn} \tag{3-52}$$

例如，当 $N=8$ 时，其周期性如图 3-6 所示。由于 $W=\mathrm{e}^{-j\frac{2\pi}{N}}=\cos\frac{2\pi}{N}-\mathrm{j}\sin\frac{2\pi}{N}$，所以，当 $N=8$ 时，可得：

$$W^N=1 \qquad W^{\frac{N}{2}}=-1$$

$$W^{\frac{N}{4}}=-\mathrm{j} \qquad W^{\frac{3N}{4}}=\mathrm{j}$$

可见，离散傅里叶变换中的乘法运算有许多重复内容。1965 年库利-图基提出把原始的 N 点序列依次分解成一系列短序列，然后，求出这些短序列的离散傅里叶变换，以此来减少乘法运算。例如，设：

$$x_1(n)=x(2n) \qquad n=0,1,2,\cdots,\frac{N}{2}-1$$

$$x_2(n)=x(2n+1) \qquad n=0,1,2,\cdots,\frac{N}{2}-1$$

由此，离散傅里叶变换可写成下面的形式：

$$\begin{aligned} X(m)&=\sum_{n=0}^{N-1}x(n)W_N^{mn} \\ &=\sum_{n=0}^{\frac{N}{2}-1}x_1(n)W_N^{mn}+\sum_{n=0}^{\frac{N}{2}-1}x_2(n)W_N^{mn} \\ &=\sum_{n=0}^{\frac{N}{2}-1}x(2n)W_N^{m(2n)}+\sum_{n=0}^{\frac{N}{2}-1}x(2n-1)W_N^{m(2n+1)} \end{aligned}$$

因为 $$W_{2N}^{k}=W_{N}^{\frac{k}{2}}$$

所以

$$\begin{aligned} X(m)&=\sum_{n=0}^{\frac{N}{2}-1}x(2n)W_{\frac{N}{2}}^{mn}+\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)W_{\frac{N}{2}}^{mn}\cdot W_N^{m} \\ &=\sum_{n=0}^{\frac{N}{2}-1}x(2n)W_{\frac{N}{2}}^{mn}+W_N^{m}\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)W_{\frac{N}{2}}^{mn} \end{aligned} \tag{3-53}$$

$$= X_1(m) + W_N^m X_2(m)$$

式中 $X_1(m)$ 和 $X_2(m)$ 分别是 $x_1(n)$ 和 $x_2(n)$ 的 $\frac{N}{2}$ 点的傅里叶变换。由于 $X_1(m)$ 和 $X_2(m)$ 均是以 $\frac{N}{2}$ 为周期，所以

$$\begin{aligned} X_1\left(m + \frac{N}{2}\right) &= X_1(m) \\ X_2\left(m + \frac{N}{2}\right) &= X_2(m) \end{aligned} \tag{3-54}$$

这说明当 $m \geqslant \frac{N}{2}$ 时，上式也是重复的。因此，

$$X(m) = X_1(m) + W_N^m X_2(m) \qquad m = 0,1,2,\cdots,N-1$$

也是成立的。

由上面的分析可见，一个 N 点的离散傅里叶变换可由两个 $N/2$ 点的傅里叶变换得到，其组合规则就是式(3-53)。离散傅里叶变换的计算时间主要由乘法决定，分解后所需乘法次数大为减少。第一项 $(N/2)^2$ 次，第二项 $(N/2)^2+N$ 次，总共为 $2\times(N/2)^2+N$ 次运算即可完成，而原来却要 N^2 次运算，可见分解后的乘法计算次数减少了近一半。当 N 是 2 的整数幂时，则上式中的 $X_1(m)$ 和 $X_2(m)$ 还可以再分成两个更短的序列，因此，计算时间会更短。由此可见，利用 W_N^m 的周期性和分解运算，从而减少乘法运算次数是实现快速运算的关键。

快速傅里叶变换简称 FFT。算法根据分解的特点一般有两类，一类是按时间分解，一类是按频率分解。下面介绍一下 FFT 的基本形式及运算蝶式流程图。

1. 基数 2 按时间分解的算法

这种算法的流程图如图 3-7 所示，图(a)输入为顺序的，运算结果是乱序的；图(b)输入为乱序的，运算结果是顺序的。上述流程图的正确性不难由公式得到的结果来验证。例如，可以由流程图(a)算得：

$$\begin{aligned} X(1) &= x_2(4) + x_2(5)W^1 \\ &= x_1(4) + x_1(6)W^2 + [x_1(5) + x_1(7)W^2]W^1 \\ &= x(0) + x(4)W^4 + [x(2) + x(6)W^4]W^2 + [x(1) + x(5)W^4]W^1 \\ &\quad + [x(3) + x(7)W^4]W^2W^1 \\ &= x(0) + x(4)W^4 + x(2)W^2 + x(6)W^4W^2 \\ &\quad + x(1)W^1 + x(5)W^4W^1 + x(3)W^2W^1 + x(7)W^4W^2W^1 \\ &= x(0) + x(4)W^4 + x(2)W^2 + x(6)W^6 + x(1)W^1 + x(5)W^5 \\ &\quad + x(3)W^3 + x(7)W^7 \end{aligned}$$

由公式计算可得如下结果：

$$X(m) = \sum_{n=0}^{N-1} x(n) W_N^{mn}$$

$$\begin{aligned} X(1) &= x(0) + x(1)W^1 + x(2)W^2 + x(3)W^3 + x(4)W^4 \\ &\quad + x(5)W^5 + x(6)W^6 + x(7)W^7 \end{aligned}$$

显然，从流程图得到的结果和利用公式得到的结果完全一致。当然，利用流程图(b)也会得到同样的结果。

2. 基数 2 按频率分解的算法

这种分解方法是直接把序列分为前 $N/2$ 点和后 $N/2$ 点两个序列，即

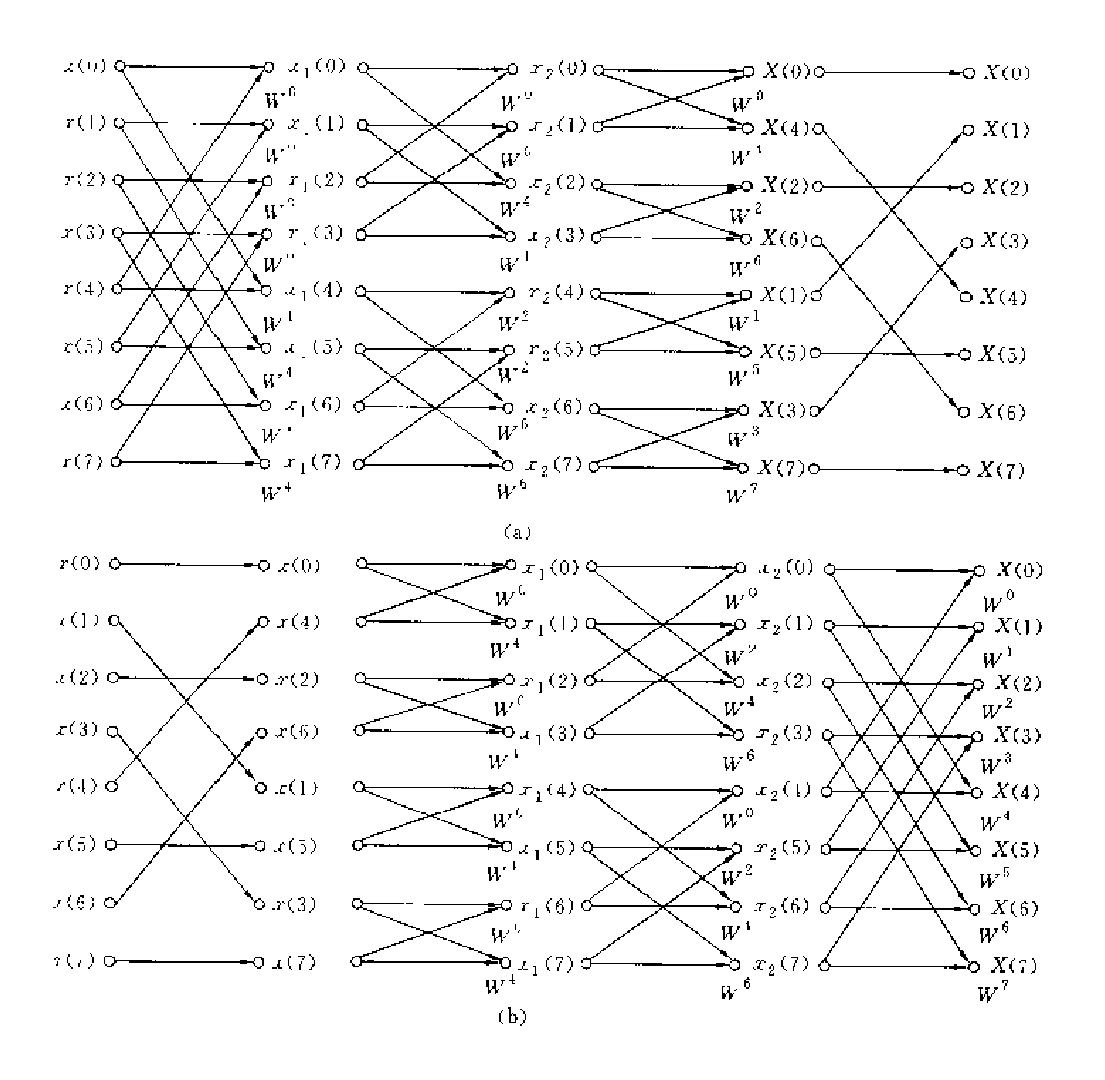


图 3-7　FFT 蝶式运算流程图（按时间分解）

$$x_1(n) = x(n) \qquad n = 0,1,2,\cdots,\frac{N}{2}-1$$

$$x_2(n) = x\left(n+\frac{N}{2}\right) \qquad n = 0,1,2,\cdots,\frac{N}{2}-1 \tag{3-55}$$

因此，离散傅里叶变换公式可写成下式：

$$\begin{aligned} X(m) &= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{mn} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{mn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x_1(n)W_N^{mn} + \sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_N^{m(n+\frac{N}{2})} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x_1(n)W_N^{mn} + (W_N^{\frac{N}{2}})^m \sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_N^{mn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) + (-1)^m x_2(n)]W_{\frac{N}{2}}^{m\frac{n}{2}} \end{aligned} \tag{3-56}$$

式中 $W_N^{\frac{N}{2}} = -1$，$W_N^{mn} = W_{\frac{N}{2}}^{m\frac{n}{2}}$，$m$ 可以分成奇数和偶数。

当 m 为偶数时，即 $m=2k$，则：

$$X(2k)=\sum_{n=0}^{\frac{N}{2}-1}[x_1(n)+x_2(n)]W_{\frac{N}{2}}^{nk} \qquad k=\frac{m}{2}=0,1,2,\cdots,\frac{N}{2}-1 \tag{3-57}$$

当 m 为奇数时，即 $m=2k+1$，则：

$$\begin{aligned}X(2k+1)&=\sum_{n=0}^{\frac{N}{2}-1}[x_1(n)-x_2(n)]W_{\frac{N}{2}}^{(2k+1)\frac{n}{2}}\\&=\sum_{n=0}^{\frac{N}{2}-1}\{[x_1(n)-x_2(n)]W_N^n\}W_{\frac{N}{2}}^{nk}\end{aligned} \tag{3-58}$$

$$k=\frac{m-1}{2}=0,1,2,\cdots,\frac{N}{2}-1$$

全部 N 点的傅里叶变换就为式(3-57)式(3-58)之和。也就是说，频率为偶数和频率为奇数的离散傅里叶变换可以分别从序列$[x_1(n)+x_2(n)]$和$[x_1(n)-x_2(n)]W_N^n$ 的$\frac{N}{2}$点傅里叶变换来求得。而每个$\frac{N}{2}$点傅里叶变换又可以由两个$\frac{N}{4}$点傅里叶变换来求得。图 3-8 为 $N=8$ 的频率分解的 FFT 流程图。仍以 $N=8$ 为例，验算如下：

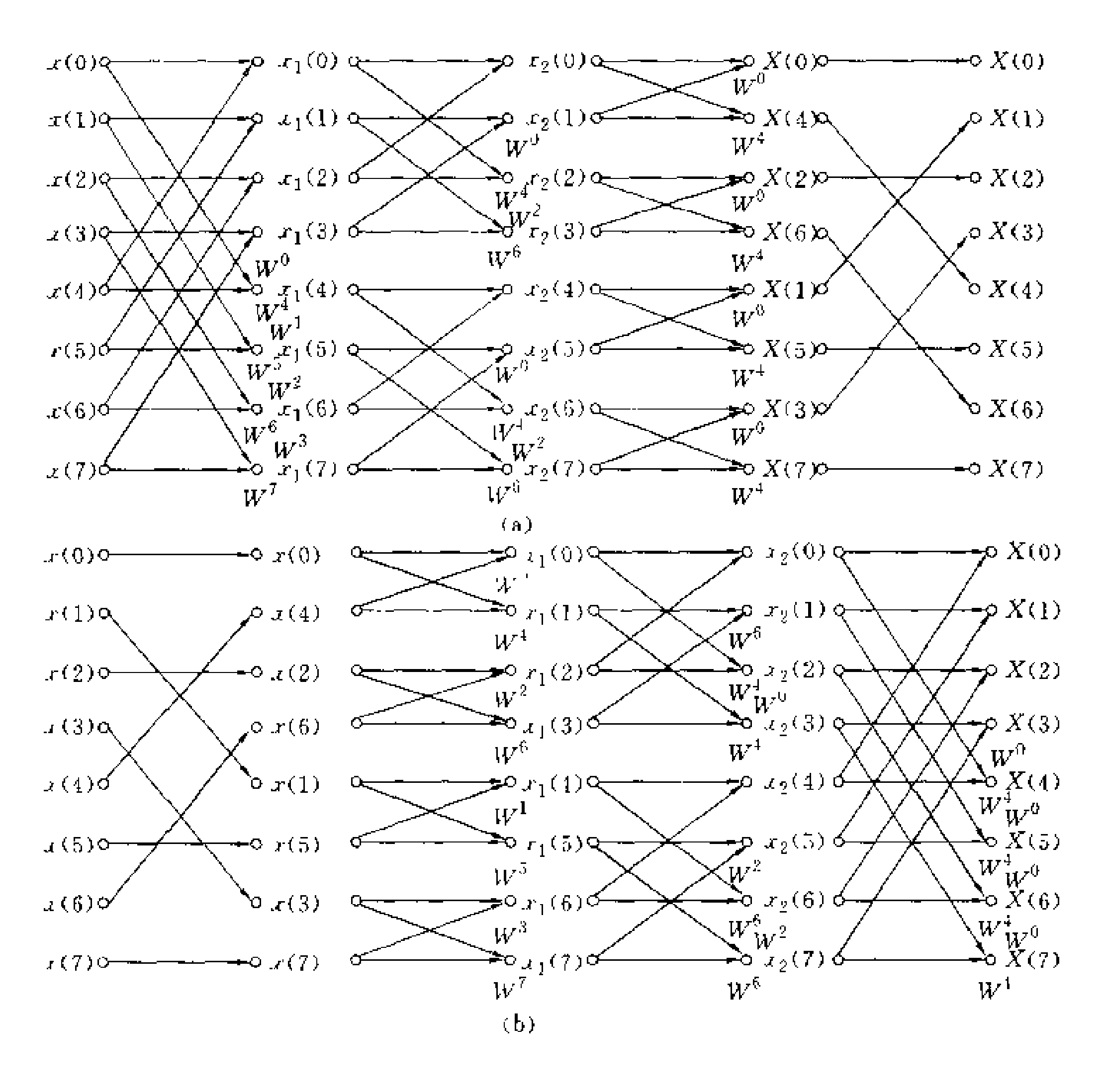


图 3-8 按频率分解 FFT 算法流程图

$$X(1)=\sum_{n=0}^{\frac{N}{2}-1}[x_1(n)-x_2(n)]W_8^n$$
$$=\sum_{n=0}^{3}[x(n)-x(n+4)]W_8^n$$
$$=[x(0)-x(4)]W_8^0+[x(1)-X(5)]W_8^1+[x(2)-x(6)]W_8^2$$
$$+[x(3)-x(7)]W_8^3$$
$$=x(0)W_8^0-x(4)W_8^0+x(1)W_8^1-x(5)W_8^1+x(2)W_8^2-x(6)W_8^2$$
$$+x(3)W_8^3-x(7)W_8^3$$

由流程图得到的结果如下，显然是一致的：

$$X(1)=x(0)W^0+x(4)W^4+x(2)W^2+x(6)W^6+x(1)W^1$$
$$+x(5)W^5+x(3)W^3+x(7)W^7$$

因为 $W^4=-W^0 \quad W^6=-W^2$

$W^5=-W^1 \quad W^7=-W^3$

所以，

$$X(1)=x(0)W^0-x(4)W^0+x(1)W^1-x(5)W^1+x(2)W^2-x(6)W^2$$
$$+x(3)W^3-x(7)W^3$$

整个快速傅里叶变换需要$\frac{N}{2}\log_2N$次复数乘法和$\frac{N}{2}\log_2N$次复数加法。当N越大时则快速算法的优越性越显著。

3.1.5 用计算机实现快速傅里叶变换

利用FFT蝶式流程图算法在计算机上实现快速傅里叶变换必须解决如下问题：迭代次数r的确定；对偶节点的计算；加权系数W_N^p的计算；重新排序问题。

(1) 迭代次数r的确定

由蝶式流程图可见，迭代次数与N有关。r值可由下式确定：

$$r=\log_2N \tag{3-59}$$

式中N是变换序列的长度。对于前述基数2的蝶式流程图N是2的整数次幂。例如，序列长度为8则要三次迭代，序列长度为16时就要4次迭代，等等。

(2) 对偶节点的计算

在流程图中把标有$x_l(k)$的点称为节点。其中下标l为列数，也就是第几次迭代，例如，$x_1(k)$则说明它是第一次迭代的结果。k代表流程图中的行数，也就是序列的序号数。其中每一节点的值均是用前一节点对计算得来的。例如，$x_1(0)$和$x_1(4)$均是$x(0)$和$x(4)$计算得来的。在蝶式流程图中，把具有相同来源的一对节点叫做对偶节点。如：$x_1(0)$和$x_1(4)$就是一对对偶节点，因为它们均来源于$x(0)$和$x(4)$。对偶节点的计算也就是求出在每次迭代中对偶节点的间隔或者节距。由流程图可见，第一次迭代的节距为$N/2$，第二次迭代的节距为$N/4$，第三次迭代的节距为$N/2^3$等等。由以上分析可得到如下对偶节点的计算方法。如果某一节点为$x_l(k)$，那么，它的对偶节点为

$$x_l\left(k-\frac{N}{2^l}\right) \tag{3-60}$$

式中l是第几次迭代的数字，k是序列的序号数，N是序列长度。

例　如果序列长度 $N=8$，求 $x_2(1)$ 的对偶节点。

利用式(3-60)计算，可得

$$x_l\left(k+\frac{N}{2^l}\right)=x_2\left(1+\frac{8}{2^2}\right)=x_2(3)$$

$$x_2(1)=x_1(1)+W_8^0x_1(3)$$

则

$$x_2(3)=x_1(1)+W_8^4x_1(3)$$

其正确性不难由流程图来验证。

(3) 加权系数 W_N^p 的计算

W_N^p 的计算主要是确定 p 值。p 值可用下述方法求得：

1) 把 k 值写成 r 位的二进制数(k 是序列的序号数，r 是迭代次数)；

2) 把这个二进制数右移 $r-1$ 位，并把左边的空位补零(结果仍为 r 位)；

3) 把这个右移后的二进制数进行比特倒转；

4) 把这比特倒转后的二进制数翻成十进制数就得到 p 值。

例：求 $x_2(2)$ 的加权系数 W_8^p。

由 $x_2(2)$ 和 W_8^p 可知 $k=2$，$l=2$，$N=8$，则

$$r=\log_2N=\log_28=3$$

1) 因为 $k=2$，所以写成二进制数为 010；

2) $r-l=3-2=1$，把 010 左移一位得到 001；

3) 把 001 做位序颠倒，即做比特倒转，得到 100；

4) 把 100 译成十进制数，得到 4，所以 $p=4$，$x_2(2)$ 的加权值为 W_8^4。

结合对偶节点的计算，可以看出 W_N^p 具有下述规律：如果某一节点上的加权系数为 W_N^p，则其对偶节点的加权系数必然是 $W_N^{p-\frac{N}{2}}$，而且 $W_N^p=-W_N^{p-\frac{N}{2}}$，所以一对对偶节点可用下式计算：

$$x_l(k)=x_{l-1}(k)+W_N^px_{l-1}\left(k+\frac{N}{2^l}\right) \tag{3-61}$$

$$x_l\left(k+\frac{N}{2^l}\right)=x_{l-1}(k)-W_N^px_{l-1}\left(k+\frac{N}{2^l}\right) \tag{3-62}$$

(4) 重新排序

由蝶式流程图可见，如果序列 $x(n)$ 是按顺序排列的，经过蝶式运算后，其变换序列是非顺序排列的，即乱序的；反之，如果 $x(n)$ 是乱序的，那么，$X(m)$ 就是顺序的。因此，为了便于输出使用，最好加入重新排序程序，以便保证 $x(n)$ 与它的变换系数 $X(m)$ 的对应关系。具体排序方法如下：

1) 将最后一次迭代结果 $x_l(k)$ 中的序号数 k 写成二进制数，即

$$x_l(k)=x_l(k_{r-1}k_{r-2}\cdots k_1k_0)$$

2) 将 r 位的二进制数比特倒转，即

$$x_l(k_0k_1\cdots k_{r-2}k_{r-1})$$

也就是 $X(m)=X(k_0k_1\cdots k_{r-2}k_{r-1})$。

3) 求出倒置后的二进制数代表的十进制数，就可以得到与 $x(k)$ 相对应的 $X(m)$ 的序号数。

例：

$$x_3(0) \to x_3(000) \to X(000) \to X(0)$$
$$x_3(1) \to x_3(001) \to X(100) \to X(4)$$
$$x_3(2) \to x_3(010) \to X(010) \to X(2)$$
$$x_3(3) \to x_3(011) \to X(110) \to X(6)$$
$$x_3(4) \to x_3(100) \to X(001) \to X(1)$$
$$x_3(5) \to x_3(101) \to X(101) \to X(5)$$
$$x_3(6) \to x_3(110) \to X(011) \to X(3)$$
$$x_3(7) \to x_3(111) \to X(111) \to X(7)$$

由蝶式流程图中可见，在中间迭代中不必考虑节点值的排序问题，在迭代运算全部完成后使用比特倒转法就可以得到正确的变换系数的顺序。

解决了上述几个关键问题之后便可以设计程序。附录 1 中给出一个 FFT 的子程序，供读者参考。

3.1.6 二维离散傅里叶变换

一幅静止的数字图像可看做是二维数据阵列。因此，数字图像处理主要是二维数据处理。二维离散傅里叶变换的定义可用下面二式表示。正变换式为

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\exp\left[-\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right] \tag{3-63}$$

$$u=0,1,2,\cdots,M-1$$
$$v=0,1,2,\cdots,N-1$$

反变换式为

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\exp\left[\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right] \tag{3-64}$$

$$x=0,1,2,\cdots,M-1$$
$$y=0,1,2,\cdots,N-1$$

在图像处理中，一般总是选择方形阵列，所以通常情况下总是 $M=N$。因此，二维离散傅里叶变换多采用下面两式形式：

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\exp\left[-\mathrm{j}2\pi\left(\frac{(ux+vy)}{N}\right)\right] \tag{3-65}$$

$$u,v=0,1,2,\cdots,N-1$$

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}F(u,v)\exp\left[\mathrm{j}2\pi\left(\frac{(ux+vy)}{N}\right)\right] \tag{3-66}$$

$$x,y=0,1,2,\cdots,N-1$$

式中符号 $F(u,v)$ 可称为空间频率。

二维离散傅里叶变换的可分离性是显而易见的：

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\exp\left(-\mathrm{j}2\pi\frac{ux}{N}\right)\times\sum_{y=0}^{N-1}f(x,y)\exp\left(-\mathrm{j}2\pi\frac{vy}{N}\right) \tag{3-67}$$

$$u,v=0,1,2,\cdots,N-1$$

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\exp\left(\mathrm{j}2\pi\frac{ux}{N}\right)\times\sum_{v=0}^{N-1}F(u,v)\exp\left(\mathrm{j}2\pi\frac{vy}{N}\right) \tag{3-68}$$

$$x,y = 0,1,2,\cdots,N-1$$

这个性质可以使二维变换用两次一维变换实现。除此之外，二维离散傅里叶变换有平移特性，即：

$$f(x,y)\exp\left[\mathrm{j}2\pi\frac{u_0x+v_0y}{N}\right] \Leftrightarrow F(u-u_0,v-v_0) \tag{3-69}$$

$$f(x-x_0,y-y_0) \Leftrightarrow F(u,v)\exp\left[-\mathrm{j}2\pi\frac{u_0x+v_0y}{N}\right] \tag{3-70}$$

周期性，即：

$$\begin{aligned} F(u,v) &= F(u+N,v) = F(u,v+N) \\ &= F(u+N,v+N) \end{aligned} \tag{3-71}$$

共轭对称性，即：

$$F(u,v) = F^*(-u,-v) \tag{3-72}$$

此外，与连续二维傅里叶变换一样，二维离散傅里叶变换也具有线性、旋转性、相关定理、卷积定理、比例性等性质。这些性质在分析及处理图像时有重要意义。例如，要观察一个完全周期的傅里叶谱，往往把变换原点移到$\left(\frac{N}{2},\frac{N}{2}\right)$处，这样才能看到整个谱图。而这个目的只要利用平移特性就可方便地达到。即

$$f(x,y)(-1)^{x+y} \Leftrightarrow F\left(u-\frac{N}{2},v-\frac{N}{2}\right) \tag{3-73}$$

用傅里叶变换分析处理图像信息有许多优点，应用也相当普遍。但是，它也有一些缺点，其一是傅里叶变换需要计算复数而不是实数，其二是收敛速度慢。因此，在有些场合下，傅里叶变换不一定是理想的变换方法。二维傅里叶变换的处理结果示于图 3-9 中。

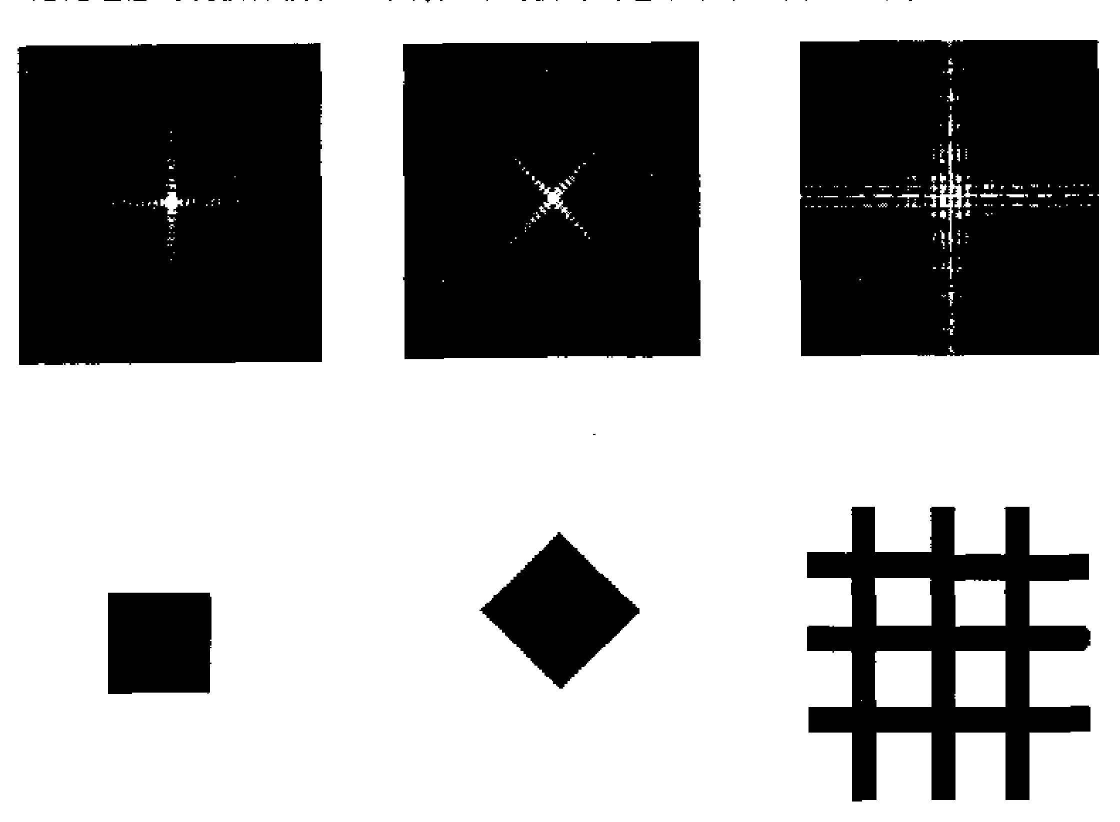


图 3-9 二维傅里叶变换的处理结果

3.2 离散余弦变换

图像处理中常用的正交变换除了傅里叶变换外，还有其他一些有用的正交变换。其中离散余弦就是一种，离散余弦变换表示为DCT。

3.2.1 离散余弦变换的定义

一维离散余弦变换的定义由下式表示：

$$F(0)=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}f(x) \tag{3-74}$$

$$F(u)=\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\frac{2(x+1)u\pi}{2N} \tag{3-75}$$

式中 $F(u)$ 是第 u 个余弦变换系数，u 是广义频率变量，$u=1,2,\cdots,N-1$；$f(x)$ 是时域 N 点序列，$x=0,1,2\cdots,N-1$。

一维离散余弦反变换由下式表示：

$$f(x)=\sqrt{\frac{1}{N}}F(0)+\sqrt{\frac{2}{N}}\sum_{u=1}^{N-1}F(u)\cos\frac{(2x-1)u\pi}{2N} \tag{3-76}$$

显然，式(3-74)式(3-75)和式(3-76)构成了一维离散余弦变换对。

二维离散余弦变换的定义由下式表示：

$$\begin{aligned}
F(0,0)&=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\\
F(0,v)&=\frac{\sqrt{2}}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cdot\cos\frac{(2y+1)v\pi}{2N}\\
F(u,0)&=\frac{\sqrt{2}}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\frac{(2x+1)u\pi}{2N}\\
F(u,v)&=\frac{2}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cdot\cos\frac{(2y+1)v\pi}{2N}
\end{aligned} \tag{3-77}$$

式(3-77)是正变换公式。其中 $f(x,y)$ 是空间域二维向量之元素；$x,y=0,1,2,\cdots,N-1$，$F(u,v)$ 是变换系数阵列之元素；式中表示的阵列为 $N\times N$。

二维离散余弦反变换由下式表示：

$$\begin{aligned}
f(x,y)=&\frac{1}{N}F(0,0)+\frac{\sqrt{2}}{N}\sum_{v=1}^{N-1}F(0,v)\cos\frac{(2y+1)v\pi}{2N}\\
&+\frac{\sqrt{2}}{N}\sum_{u=1}^{N-1}F(u,0)\cos\frac{(2x+1)u\pi}{2N}\\
&+\frac{2}{N}\sum_{u=1}^{N-1}\sum_{v=1}^{N-1}F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cdot\cos\frac{(2y+1)v\pi}{2N}
\end{aligned} \tag{3-78}$$

式中的符号意义同正变换式一样。式(3-77)和式(3-78)是离散余弦变换的解析式定义。更为简洁的定义方法是采用矩阵式定义。如果令 $N=4$，由一维解析式定义可得如下展开式：

$$
\begin{cases}
F(0) = 0.500f(0) + 0.500f(1) + 0.500f(2) + 0.500f(3) \\
F(1) = 0.653f(0) + 0.271f(1) - 0.271f(2) - 0.653f(3) \\
F(2) = 0.500f(0) - 0.500f(1) - 0.500f(2) + 0.500f(3) \\
F(3) = 0.271f(0) - 0.653f(1) + 0.653f(2) - 0.271f(3)
\end{cases}
\tag{3-79}
$$

写成矩阵式：

$$
\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix} = \begin{bmatrix} 0.500 & 0.500 & 0.500 & 0.500 \\ 0.653 & 0.271 & -0.271 & -0.653 \\ 0.500 & -0.500 & -0.500 & 0.500 \\ 0.271 & -0.653 & 0.653 & -0.271 \end{bmatrix} \cdot \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix}
\tag{3-80}
$$

若定义$[A]$为变换矩阵，$[F(u)]$为变换系数矩阵，$[f(x)]$为时域数据矩阵，则一维离散余弦变换的矩阵定义式可写成如下形式：

$$
[F(u)] = [A][f(x)]
\tag{3-81}
$$

同理，可得到反变换展开式：

$$
\begin{cases}
f(0) = 0.500F(0) + 0.653F(1) + 0.500F(2) + 0.271F(3) \\
f(1) = 0.500F(0) + 0.271F(1) - 0.500F(2) - 0.653F(3) \\
f(2) = 0.500F(0) - 0.271F(1) - 0.500F(2) + 0.653F(3) \\
F(3) = 0.500F(0) - 0.653F(1) + 0.500F(2) - 0.271F(3)
\end{cases}
\tag{3-82}
$$

写成矩阵式：

$$
\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} = \begin{bmatrix} 0.500 & 0.653 & 0.500 & 0.271 \\ 0.500 & 0.271 & -0.500 & -0.653 \\ 0.500 & -0.271 & -0.500 & 0.653 \\ 0.500 & -0.653 & 0.500 & -0.271 \end{bmatrix} \cdot \begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix}
\tag{3-83}
$$

即

$$
[f(x)] = [A]^{T}[F(u)]
\tag{3-84}
$$

当然，二维离散余弦变换也可以写成矩阵式：

$$
\begin{aligned}
&[F(u,v)] = [A][f(x,y)][A]^{T} \\
&[f(x,y)] = [A]^{T}[F(u,v)][A]
\end{aligned}
\tag{3-85}
$$

式中$[f(x,y)]$是空间数据阵列，$[F(u,v)]$是变换系数阵列，$[A]$是变换矩阵，$[A]^{T}$是$[A]$的转置。

3.2.2 离散余弦变换的正交性

由一维 DCT 的定义可知：

$$
F(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x)
$$

$$
F(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N}
$$

它的基向量是

$$
\left\{ \sqrt{\frac{1}{N}}, \sqrt{\frac{2}{N}} \cos \frac{(2x+1)u\pi}{2N} \right\}
\tag{3-86}
$$

在高等数学中，切比雪夫多项式的定义为

$$T_0(p) = \sqrt{\frac{1}{N}}$$

$$T_u(z_x) = \sqrt{\frac{2}{N}}\cos[u\arccos(z_x)] \tag{3-87}$$

式中 $T_u(z_x)$是 u 和 z_x 的多项式。它的第 N 个多项式为

$$T_N(z_x) = \sqrt{\frac{2}{N}}\cos[N\arccos(z_x)]$$

如果 $$T_N(z_x) = 0$$

那么，$$z_x = \cos\frac{(2x+1)\pi}{2N}$$

将此式代入 $T_u(z_x)$，则

$$T_u = \sqrt{\frac{2}{N}}\cos\left\{u\arccos\left[\cos\frac{(2x+1)\pi}{2N}\right]\right\}$$

$$= \sqrt{\frac{2}{N}}\cos\frac{(2x+1)u\pi}{2N} \tag{3-88}$$

显然，这与一维 DCT 的基向量是一致的。因为切比雪夫多项式是正交的，所以 DCT 也是正交的。另外，离散余弦变换的正交性也可以通过实例看出。如前所示，当 $N=4$ 时，

$$[A] = \begin{bmatrix} 0.500 & 0.500 & 0.500 & 0.500 \\ 0.653 & 0.271 & -0.271 & -0.653 \\ 0.500 & -0.500 & -0.500 & 0.500 \\ 0.271 & -0.653 & 0.653 & -0.271 \end{bmatrix}$$

$$[A]^{\mathrm{T}} = \begin{bmatrix} 0.500 & 0.635 & 0.500 & 0.2710 \\ 0.500 & 0.271 & -0.500 & -0.653 \\ 0.500 & -0.271 & -0.500 & 0.653 \\ 0.500 & -0.653 & 0.500 & -0.271 \end{bmatrix}$$

显然$[A][A]^{\mathrm{T}}=[I]$，这是满足正交条件的。从上述讨论可见，离散余弦变换是一类正交变换。

3.2.3 离散余弦变换的计算

与傅里叶变换一样，离散余弦变换自然可以由定义式出发进行计算。但这样的计算量太大，在实际应用中很不方便。所以也要寻求一种快速算法。

首先，从定义出发，作如下推导：

$$F(u) = \sqrt{\frac{2}{N}}\sum_{x=0}^{N-1} f(x)\cos\frac{(2x+1)u\pi}{2N}$$

$$= \sqrt{\frac{2}{N}}\sum_{x=0}^{N-1} f(x)\mathrm{Re}\left\{e^{-j\frac{(2x+1)u\pi}{2N}}\right\} \tag{3-89}$$

$$= \sqrt{\frac{2}{N}}\mathrm{Re}\left\{\sum_{x=0}^{N-1} f(x)e^{-j\frac{(2x+1)u\pi}{2N}}\right\}$$

式中 Re 是取其实部的意思。如果把时域数据向量作下列延拓，即：

$$f_e(x)=\begin{cases}f(x) & x=0,1,2,\cdots,N-1\\ 0 & x=N,N+1,\cdots\cdots,2N-1\end{cases} \tag{3-90}$$

则 $f_e(x)$的离散余弦变换可写成下式：

$$\begin{aligned}F(0)&=\frac{1}{\sqrt{N}}\sum_{x=0}^{2N-1}f_e(x)\\ F(u)&=\sqrt{\frac{2}{N}}\sum_{x=0}^{2N-1}f_e(x)\cos\frac{(2x+1)u\pi}{2N}\\ &=\sqrt{\frac{2}{N}}R_e\left\{\sum_{x=0}^{2N-1}f_e(x)\mathrm{e}^{-\mathrm{j}\frac{(2x+1)u\pi}{2N}}\right\}\\ &=\sqrt{\frac{2}{N}}R_e\left\{\mathrm{e}^{-\mathrm{j}\frac{u\pi}{2N}}\cdot\sum_{x=0}^{2N-1}f_e(x)\mathrm{e}^{-\mathrm{j}\frac{2xu\pi}{2N}}\right\}\end{aligned} \tag{3-91}$$

由式(3-91)可见，

$$\sum_{x=0}^{2N-1}f_e(x)\mathrm{e}^{-\mathrm{j}\frac{2xu\pi}{2N}}$$

是 $2N$ 点的离散傅里叶变换。所以，在作离散余弦变换时，可以把序列长度延拓为 $2N$，然后作离散傅里叶变换，产生的结果取其实部便可得到余弦变换。

同理，在作反变换时，首先在变换空间，把$[F(u)]$作如下延拓：

$$F_e(u)=\begin{cases}F(u) & u=0,1,2,\cdots,N-1\\ 0 & u=N,N+1,\cdots,2N-1\end{cases} \tag{3-92}$$

那么，反变换也可用式(3-93)表示：

$$\begin{aligned}f(x)&=\frac{1}{\sqrt{N}}F_e(0)+\sqrt{\frac{2}{N}}\sum_{u=1}^{2N-1}F_e(u)\cos\frac{(2x+1)u\pi}{2N}\\ &=\frac{1}{\sqrt{N}}F_e(0)+\sqrt{\frac{2}{N}}\sum_{u=1}^{2N-1}F_e(u)\mathrm{Re}\left\{\mathrm{e}^{\mathrm{j}\frac{(2x+1)u\pi}{2N}}\right\}\\ &=\frac{1}{\sqrt{N}}F_e(0)+\sqrt{\frac{2}{N}}\sum_{u=1}^{2N-1}F_e(u)\mathrm{Re}\left\{\mathrm{e}^{\mathrm{j}\frac{2xu\pi}{2N}}\cdot\mathrm{e}^{\mathrm{j}\frac{u\pi}{2N}}\right\}\\ &=\frac{1}{\sqrt{N}}F_e(0)+\sqrt{\frac{2}{N}}\mathrm{Re}\left\{\sum_{u=1}^{2N-1}F_e(u)\cdot\mathrm{e}^{\mathrm{j}\frac{u\pi}{2N}}\cdot\mathrm{e}^{\mathrm{j}\frac{2xu\pi}{2N}}\right\}\\ &=\left[\frac{1}{\sqrt{N}}-\sqrt{\frac{2}{N}}\right]F_e(0)+\sqrt{\frac{2}{N}}\mathrm{Re}\left\{\sum_{u=0}^{2N-1}\left[F_e(u)\cdot\mathrm{e}^{\mathrm{j}\frac{u\pi}{2N}}\right]\mathrm{e}^{\mathrm{j}\frac{2xu\pi}{2N}}\right\}\end{aligned} \tag{3-93}$$

由式(3-93)可见，离散余弦反变换可以从$\left[F_e(u)\cdot\mathrm{e}^{\mathrm{j}\frac{u\pi}{2N}}\right]$的 $2N$ 点反傅里叶变换实现。

通过快速离散余弦变换的原理分析，则不难用计算机实现快速余弦变换。附录 2 中给出 $2N$ 点 FFT 实现快速 DCT 的程序，供读者参考。

3.3 沃尔什变换

离散傅里叶变换和余弦变换在快速算法中都要用到复数乘法，占用的时间仍然比较多。在某些应用领域中，需要更为便利更为有效的变换方法，沃尔什变换就是其中的一种。

沃尔什函数是在1923年由美国数学家沃尔什(Walsh)提出来的。在沃尔什的原始论文中，给出了沃尔什函数的递推公式，这个公式是按照函数的序数由正交区间内过零点平均数来定义的。不久以后，这种规定函数序数的方法也被波兰数学家卡兹马兹(Kaczmarz)采用了，所以，通常将这种规定函数序数的方法称为沃尔什-卡兹马兹(Walsh-Kaczmarz)定序法。

1931年，美国数学家佩利(Paley)又给沃尔什函数提出了一个新的定义。他指出，沃尔什函数可以用有限个拉德梅克(Rademacher)函数的乘积来表示。这样得到的函数的序数与沃尔什得到的函数的序数完全不同，这种定序方法是用二进制来定序的，所以称为二进制序数或自然序数。

利用只包含+1和-1阵元的正交矩阵可以将沃尔什函数表示为矩阵形式。早在1867年，英国数学家希尔威斯特(Sylvester)已经研究过这种矩阵。后来，法国数学家哈达玛(Hadamard)在1893年将这种矩阵加以普遍化，建立了所谓哈达玛矩阵。利用克罗内克乘积算子(Kronecker Product Operator)不难把沃尔什函数表示为哈达玛矩阵形式。利用这种形式定义的沃尔什函数称为克罗内克序数。这就是沃尔什函数的第三种定序法。

由上述历史可见，沃尔什函数及其有关函数的数学基础早已奠定了。但是，这些函数在工程中得到应用却是近几十年的事情。主要原因是由于半导体器件和计算机在近几十年得到迅速发展，它们的发展为沃尔什函数的实用解决了手段问题，因此，也使沃尔什函数得到了进一步发展。与傅里叶变换相比，沃尔什变换的主要优点在于存储空间少和运算速度高，这一点对图像处理来说是至关重要的，特别是在大量数据需要进行实时处理时，沃尔什函数就更加显示出它的优越性。

3.3.1 正交函数的概念

一组实值的连续函数$\{S_n(t)\}=\{S_0(t),S_1(t),S_2(t),\cdots\}$，在$0\leqslant t\leqslant T$区间内，如果满足下式：

$$\int_0^T kS_n(t)\cdot S_m(t)\mathrm{d}t=\begin{cases}k & m=n\\ 0 & m\neq n\end{cases} \tag{3-94}$$

叫$\{S_n(t)\}$在区间$0\leqslant t\leqslant T$内是正交的。m、n是正实数，k是与m、n无关的非负常数。如果$k=1$，称为归一化正交。任一组非归一化的正交函数总可以变换为归一化正交函数。

如果$f(t)$是定义在$(0,T)$区间上的实值信号，利用正交函数可表示为下式：

$$f(t)=\sum_{n=0}^{\infty}C_nS_n(t) \tag{3-95}$$

式中C_n是第n项系数。

一组完备的正交函数必然是一组闭合函数。完备的正交函数组必须满足下述两个条件。第一，不存在这样一个函数$x(t)$，它满足：

$$0<\int_T x^2(t)\mathrm{d}t<\infty \tag{3-96}$$

而且也满足：

$$\int_T x(t)S_n(t)\mathrm{d}t=0 \qquad n=0,1,2,\cdots \tag{3-97}$$

这个条件的意思是说，再也没有不属于$\{S_n(t)\}$的某个非零的函数$x(t)$，它与$\{S_n(t)\}$的每一个函数正交。也就是说，所有互相正交的函数都包括在$\{S_n(t)\}$里面。

第二，对于任何满足 $\int_T f^2(t)\mathrm{d}t < \infty$ 的函数，对于给定的任意微小正数 $\varepsilon>0$，总存在一个正整数 N 与有限展开式：

$$\hat{f}(t) = \sum_{n=0}^{N-1} C_n S_n(t) \tag{3-98}$$

使得

$$\int_T |f(t) - \hat{f}(t)|^2 \mathrm{d}t < \varepsilon \tag{3-99}$$

这一条件的意思是，任一个能量有限的信号 $f(t)$ 总可以用有限级数来逼近它。对于给定的误差来说，总可以找到一个 N 值，使这种逼近的精确度满足要求。

完备性的必要和充分条件是在正交区间内各分量函数的平方之和存在，并且应该完全满足帕斯维尔定理。这个条件的物理意义是：一组完备的正交函数所包含的能量，无论是在时域中还是在变换域中都是相同的。完备性的重要意义在于：只有当正交函数是完备的，我们才能将一个满足一定条件的函数展开成此正交函数系的级数，否则将不能保证能量相等。

正交性、完备性、归一化的定义适用于一组函数中的所有函数。它所规定的区间可以是半无限区间 $(0,+\infty)$，也可以是全无限区间 $(-\infty,+\infty)$，当然也可以是有限区间 $\left(-\frac{T}{2},+\frac{T}{2}\right)$，$(0,T)$ 等。

沃尔什函数的一个有用的特点是由有限多个沃尔什函数组成的时间受限信号在变换域中只占据有限区间。而对于圆函数来说则不然，一般来说，时间受限，其频谱区间则是无限的，反之，频域受限信号，在时间域上则是无限的。

3.3.2 拉德梅克函数

拉德梅克(Rademacher)函数集是一个不完备的正交函数集，由它可以构成完备的沃尔什函数。在这里首先介绍一下拉德梅克函数。拉德梅克函数包括 n 和 t 两个自变量，用 $R(n,t)$ 来表示拉德梅克函数。把一个正弦函数作无限限幅就可以得到拉德梅克函数。它可用下式来表示：

$$R(n,t) = \mathrm{sgn}(\sin 2^n \pi t) \tag{3-100}$$

$$\mathrm{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases} \tag{3-101}$$

当 $x=0$ 时，$\mathrm{sgn}(x)$ 无定义。

由 sin 函数的周期性知道 $R(n,t)$ 也是周期性函数。由式(3-100)可见，当 $n=1$ 时，$R(n,t)$ 的周期为 1；$n=2$ 时 $R(2,t)$ 的周期为 $\frac{1}{2}$；当 $n=3$ 时，$R(3,t)$ 的周期为 $\frac{1}{2^2}$；一般情况下可用下式表示：

$$R(n,t) = R\left(n, t - \frac{1}{2^{n-1}}\right) \qquad n = 1,2,\cdots \tag{3-102}$$

拉德梅克函数的波形如图 3-10 所示。由图 3-10 可见，拉德梅克函数有如下规律：

1) $R(n,t)$ 的取值只有 $+1$ 和 -1。

2) $R(n,t)$ 是 $R(n-1,t)$ 的二倍频。因此，如果已知最高次数 $m=n$，则其他拉德梅克函数可由脉冲分频器来产生。

3) 如果已知 n，那么 $R(n,t)$ 有 2^{n-1} 个周期，其中 $0<t<1$；如果在 $t=\frac{k+\frac{1}{2}}{2^n}$ 处作取样，则可

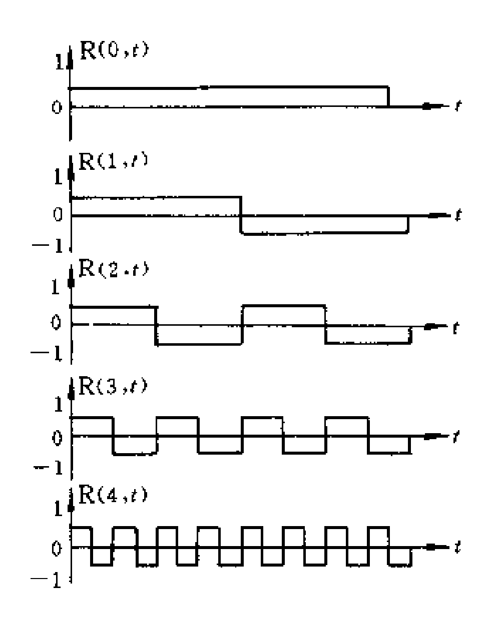


图 3-10　拉德梅克函数

得到一数据序列 $R(n,k)$，$k=0,1,2,\cdots,2^{n-1}$。每一取样序列将与下述矩阵相对应。这里我们取 $n=3$，$k=0,1,2,\cdots,7$。

$$\begin{bmatrix} R(0,k) \\ R(1,k) \\ R(2,k) \\ R(3,k) \end{bmatrix} \Leftrightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{3-103}$$

采用上述离散矩阵形式就可以用计算机进行灵活处理。

3.3.3　沃尔什函数

沃尔什函数系是完备的正交函数系，其值也是只取+1和−1。从排列次序来定义不外乎三种：一种是按沃尔什排列或称按列率排列来定义；第二种是按佩利排列或称自然排列来定义；第三种是按哈达玛排列来定义。还可用其他方式来定义，但沃尔什函数的定义至今尚未统一，下面分别讨论上述三种排列方法定义的沃尔什函数。

1. 按沃尔什排列的沃尔什函数

按沃尔什排列的沃尔什函数用 $\mathrm{wal_W}(i,t)$来表示。函数波形如图 3-11 所示。

按沃尔什排列的沃尔什函数实际上就是按列率排列的沃尔什函数。通常把正交区间内波形变号次数的 1/2 称为列率(sequency)。如果令 i 为波形在正交区间内的变号次数，那么，按照 i 为奇数或偶数，函数 $\mathrm{wal_W}(i,t)$的列率将分别由下式来决定：

$$S_i = \begin{cases} 0 & i=0 \\ \dfrac{i+1}{2} & i=\mathrm{odd} \\ \dfrac{i}{2} & i=\mathrm{even} \end{cases} \tag{3-104}$$

按沃尔什排列的沃尔什函数可由拉德梅克函数构成，它的表达式如下：

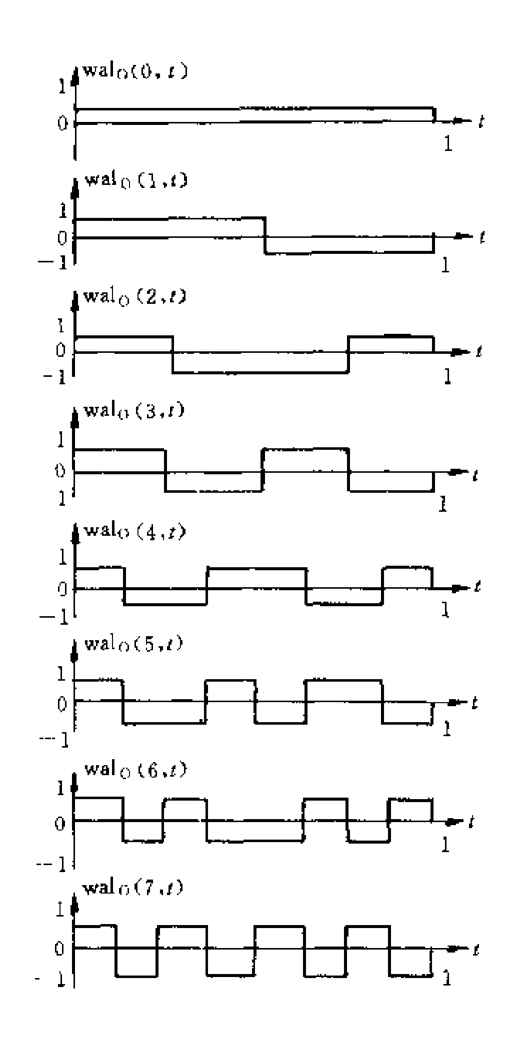


图 3-11　按沃尔什排列的沃尔什函数

$$\mathrm{wal}_{\mathrm{W}}(i,t)=\prod_{k=0}^{p-1}[R(k+1,t)]^{g(i)_k}\qquad g(i)_k\in\{0,1\}\tag{3-105}$$

式中 $R(k+1,t)$是拉德梅克函数，$g(i)$是 i 的格雷码，$g(i)_k$ 是此格雷码的第 k 位数字，p 为正整数。

一个正整数可以编成自然二进码，但也可以编成格雷码。格雷码也称为反射码。格雷码的特点是：两个相邻数的格雷码只有一个码位的值不同。例如，2 的格雷码是(0011)，3 的格雷码为(0010)。这两个相邻的数字的格雷码只有第 4 个码位的值不同。在脉冲编码技术中，常常采用这种码，以便得到较好的误差特性。一个正整数的自然二进码和格雷码之间是可以互相转换的。从自然二进码转成格雷码的方法如下：

设一个十进制数的自然二进码为

$$n=(n_{p-1}n_{p-2}\cdots n_k\cdots n_2n_1n_0)_{\mathrm{B}}$$

并设 n 的格雷码为

$$g=(g_{p-1}g_{p-2}\cdots g_k\cdots g_2g_1g_0)_{\mathrm{G}}$$

其中 n_k 和 g_k 分别为二进码和格雷码内的码位数字，并且 n_k、$g_k\in\{0,1\}$。它们之间的关系可用式(3-106)表示：

则

$$
\begin{cases}
g_{p-1} = n_{p-1} \\
g_{p-2} = n_{p-1} \oplus n_{p-2} \\
g_{p-3} = n_{p-2} \oplus n_{p-3} \\
\quad \cdots \\
g_k = n_{k-1} \oplus n_k \\
\quad \cdots \\
g_1 = n_2 \oplus n_1 \\
g_0 = n_1 \oplus n_0
\end{cases}
\tag{3-106}
$$

式中，$\oplus$代表模 2 加。

例　试求(2)的格雷码。

解：　$\because \quad n(2)=(0010)_B$

其中，

$$n_3 = 0$$
$$n_2 = 0$$
$$n_1 = 1$$
$$n_0 = 0$$

$$\therefore \quad g_3 = n_3 = 0$$
$$g_2 = n_3 \ominus n_2 = 0 \oplus 0 = 0$$
$$g_1 = n_2 \oplus n_1 = 0 \oplus 1 = 1$$
$$g_0 = n_1 \oplus n_0 = 1 \oplus 0 = 1$$

其格雷码　$g(2) = (0011)_G$

同理，若 $n(3)=(0011)_B$，则其格雷码为 $g(3)=(0010)_G$。

在格雷码中，有如下关系存在：

$$g(m) \oplus g(n) = g(m \oplus n) \tag{3-107}$$

例　设：

$$m = 2 = (0010)_B$$
$$n = 3 = (0011)_B$$
$$g(2) = (0011)_G$$
$$g(3) = (0010)_G$$

则　$g(2) \oplus g(3) = (0001)$

而　$(2)_B \oplus (3)_B = (0001)$

所以，　$g(2) \oplus g(3) = g(2 \oplus 3)$

从正整数的格雷码也可以求出该十进数的自然二进码。其转换方法如下：

设正整数的格雷码为

$$g(n) = (g_{p-1} g_{p-2} g_{p-3} \cdots g_k \cdots g_2 g_1 g_0)$$

又设其自然二进码为

$$B(n) = (n_{p-1} n_{p-2} n_{p-3} \cdots n_k \cdots n_2 n_1 n_0)$$

则

$$\begin{cases} n_{p-1} = g_{p-1} \\ n_{p-2} = g_{p-1} \oplus g_{p-2} \\ n_{p-3} = g_{p-1} \oplus g_{p-2} \oplus g_{p-3} \\ \cdots \\ n_k = g_{p-1} \oplus g_{p-2} \oplus g_{p-3} \oplus \cdots \oplus g_k \\ \cdots \\ n_2 = g_{p-1} \oplus g_{p-2} \oplus g_{p-3} \oplus \cdots \oplus g_2 \\ n_1 = g_{p-1} \oplus g_{p-2} \oplus g_{p-3} \oplus \cdots \oplus g_2 \oplus g_1 \\ n_0 = g_{p-1} \oplus g_{p-2} \oplus g_{p-3} \oplus \cdots \oplus g_2 \oplus g_1 \oplus g_0 \end{cases} \tag{3-108}$$

例　n 的格雷码为 1011，求其二进码表示。

由给定的格雷码可知：　　　　$(n)_G=(1011)$

其中，

$$g_3 = 1$$
$$g_2 = 0$$
$$g_1 = 1$$
$$g_0 = 1$$

所以，

$$n_3 = g_3 = 1$$
$$n_2 = g_3 \oplus g_2 = 1 \oplus 0 = 1$$
$$n_1 = g_3 \oplus g_2 \oplus g_1 = 1 \oplus 0 \oplus 1 = 0$$
$$n_0 = g_3 \oplus g_2 \oplus g_1 \oplus g_0 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

即：二进码为$(1101)_B$

以上便是格雷码的定义及格雷码与自然二进码之间的转换方法。下面我们再回过头来看由拉德梅克函数定义的按沃尔什排列的沃尔什函数。

例　用公式(3-105)求 $p=4$ 时的 $\mathrm{wal_W}(5,t)$。

解：因为 $i=5$，所以 5 的自然二进码为(0101)。由前面所述的转换规则可得到格雷码为(0111)。因此，有下面的对应关系：

(0	1	1	1)
↑	↑	↑	↑
第 3 位	第 2 位	第 1 位	第 0 位
$g(5)_3$	$g(5)_2$	$g(5)_1$	$g(5)_0$

即

$$g(5)_0 = 1,\ g(5)_1 = 1$$
$$g(5)_2 = 1,\ g(5)_3 = 0$$

代入式(3-105)得：

$$\mathrm{wal_W}(i,t) = \prod_{k=0}^{p-1} [R(k+1,t)]^{g(i)_k}$$

$$\begin{aligned} \mathrm{wal_W}(5,t) &= [R(1,t)]^1 \cdot [R(2,t)]^1 \cdot [R(3,t)]^1 \cdot [R(4,t)]^0 \\ &= R(1,t) \cdot R(2,t) \cdot R(3,t) \end{aligned}$$

例　令 $p=4$，$i=9$，求 $\text{wal}_W(9,t)$。

解：因为 $i=9$，所以其格雷码为(1101)，因此：

$$g(9)_3 = 1$$
$$g(9)_2 = 1$$
$$g(9)_1 = 0$$
$$g(9)_0 = 1$$

代入式(3-105)得：

$$\begin{aligned}\text{wal}_W(9,t) &= [R(1,t)]^1 \cdot [R(2,t)]^0 \cdot [R(3,t)]^1 \cdot [R(4,t)]^1 \\ &= R(1,t) \cdot R(3,t) \cdot R(4,t)\end{aligned}$$

摹仿正、余弦函数的奇偶对称性，按沃尔什排列的沃尔什函数也可以分成 $\text{cal}(i,t)$和 $\text{sal}(i,t)$。当 i 是偶数时称为 $\text{cal}(i,t)$，当 i 是奇数时称为 $\text{sal}(i,t)$。例如 $p=4$ 时，

$$\text{cal}(0,t) = \text{wal}_W(0,t)$$
$$\text{sal}(1,t) = \text{wal}_W(1,t)$$
$$\text{cal}(1,t) = \text{wal}_W(2,t)$$
$$\text{sal}(2,t) = \text{wal}_W(3,t)$$
$$\text{cal}(2,t) = \text{wal}_W(4,t)$$
$$\text{sal}(3,t) = \text{wal}_W(5,t)$$
$$\text{cal}(3,t) = \text{wal}_W(6,t)$$
$$\text{sal}(4,t) = \text{wal}_W(7,t)$$

即：

$$\text{wal}_W(2i,t) = \text{cal}(i,t) \qquad \text{(偶函数)}$$
$$\text{wal}_W(2i-1,t) = \text{sal}(i,t) \qquad \text{(奇函数)}$$

按沃尔什排列的沃尔什函数也可以用三角函数来定义。在正交区间[0,1]内，$i=0,1,2,\cdots,\leqslant(2^p-1)$，$p$ 为正整数的情况下，可由下式来定义：

$$\text{wal}_W(i,t) = \prod_{k=0}^{p-1} \text{sgn}[\cos i_k 2^k \pi t] \tag{3-109}$$

式中 i_k 是 i 的二进码的第 k 位数字，$i_k \in \{0,1\}$。

例　$p=3$ 时，求 $\text{wal}_W(3,t)$的三角函数表达式。

因为 $i=3$，所以其自然二进码为(011)。将 $i_2=0$，$i_1=1$，$i_0=1$，代入(3-109)式得：

$$\begin{aligned}\text{wal}_W(3,t) &= \text{sgn}[\cos 1 \times 2^0 \pi t] \cdot \text{sgn}[\cos 1 \times 2^1 \pi t] \cdot \text{sgn}[\cos 0 \times 2^2 \pi t] \\ &= \text{sgn}[\cos \pi t] \cdot \text{sgn}[\cos 2\pi t]\end{aligned}$$

当 $p=3$ 时，对前 8 个 $\text{wal}_W(i,t)$取样，同样可以写成矩阵式如下：

$$\boldsymbol{H}_W(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{3-110}$$

2. 按佩利排列的沃尔什函数

用 $\mathrm{wal_P}(i,t)$来表示按佩利排列的沃尔什函数，其波形如图 3-12 所示。按佩利排列的沃尔什函数也可以由拉德梅克函数产生。其定义由式(3-111)表示：

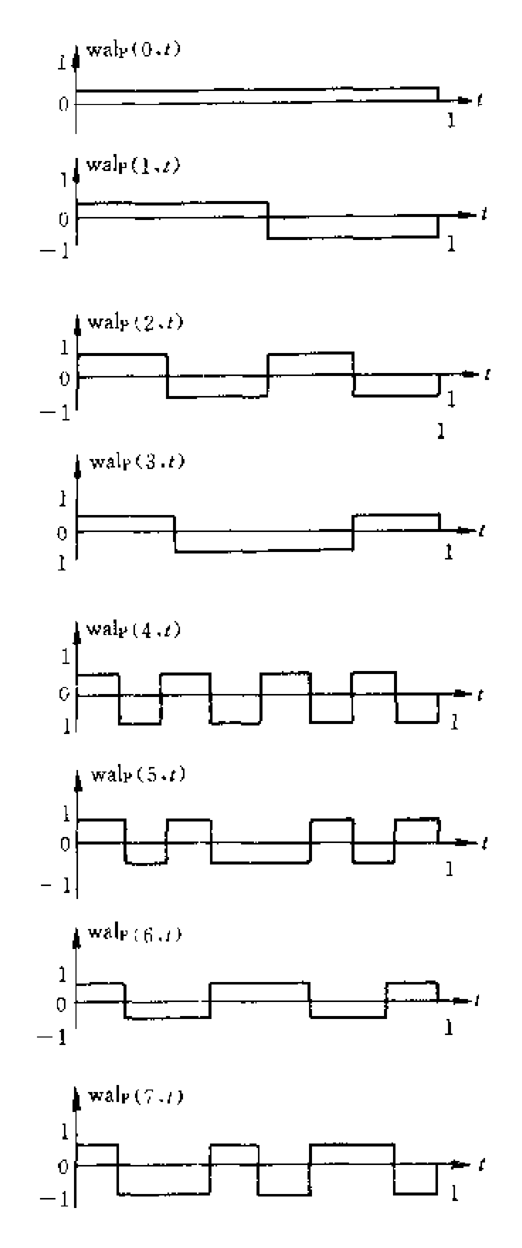


图 3-12　按佩利排列的沃尔什函数

$$\mathrm{wal_P}(i,t) = \prod_{k=0}^{p-1}[R(k+1,t)]^{i_k} \tag{3-111}$$

式中 $R(k+1,t)$是拉德梅克函数，i_k 是将函数序号写成自然二进码的第 k 位数字，$i_k\in\{0,1\}$。即：

$$(i) = (i_{n-1}i_{n-2}\cdots i_2 i_1 i_0)_B$$

例　$p=3$ 时，求 $\mathrm{wal_P}(1,t)$。

解：因为 $i=1$，所以自然二进码为

$$\begin{matrix} [0 & 0 & 1] \\ \uparrow & \uparrow & \uparrow \\ \text{第 2 位} & \text{第 1 位} & \text{第 0 位} \end{matrix}$$

代入式(3-111)得：

$$\begin{aligned} \text{wal}_p(1,t) &= \prod_{k=0}^{2}[R(k+1,t)]^{i_k} \\ &= [R(1,t)]^1 \cdot [R(2,t)]^0 \cdot [R(3,t)]^0 \\ &= R(1,t) \end{aligned}$$

例　$p=3$，求 $\text{wal}_P(5,t)$。

解：因为 $i=5$，所以，二进码为(101)，代入式(3-111)，则

$$\begin{aligned} \text{wal}_P(5,t) &= \prod_{k=0}^{2}[R(k+1,t)]^{i_k} \\ &= [R(1,t)]^1 \cdot [R(2,t)]^0 \cdot [R(3,t)0]^1 \\ &= R(1,t) \cdot R(3,t) \end{aligned}$$

当 $p=3$ 时的 8 个沃尔什函数经取样后可得矩阵 $\boldsymbol{H}_P(3)$ 如式(3-112)：

$$\boldsymbol{H}_P(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \tag{3-112}$$

由按佩利排列的沃尔什函数前 8 个波形可以看出有如下一些规律：

1）函数序号 i 与正交区间内取值符号变化次数有表 3-1 所列之关系。

表 3-1　按佩利排列的沃尔什函数序号与变号次数的关系

i	0	1	2	3	4	5	6	7
变号数	0	1	3	2	7	6	4	5

2）i 与变号次数的关系是二进码与格雷码的关系。如 $i=6=(110)_B$，这个二进制码按格雷码读出是 4，也就是说，把十进制数 i 编成自然二进码，然后按格雷码的规律反变回十进制数，这个数就是这个序号的沃尔什函数的变号次数。

3. 按哈达玛排列的沃尔什函数

按哈达玛排列的沃尔什函数是从 2^n 阶哈达玛矩阵得来的。2^n 阶哈达玛矩阵每一行的符号变化规律，对应某个沃尔什函数在正交区间内符号变化的规律，也就是说，2^n 阶哈达玛矩阵的每一行就对应一个离散沃尔什函数。2^n 阶哈达玛矩阵有如下形式：

$$H(0) = [1] \tag{3-113}$$

$$H(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3-114}$$

$$H(2)=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix} \tag{3-115}$$

$$\cdots$$

一般情况，

$$H(m)=\begin{bmatrix}H(m-1) & H(m-1)\\ H(m-1) & -H(m-1)\end{bmatrix}=H(m-1)\otimes H(1) \tag{3-116}$$

式(3-116)是哈达玛矩阵的递推关系式。利用这个关系式可以产生任意 2^n 阶哈达玛矩阵。这个关系也叫做克罗内克积(Kronecker Product)关系，或叫直积关系。

按哈达玛排列的沃尔什函数用 $\mathrm{wal}_H(i,t)$ 来表示。它的前 8 个函数波形如图 3-13 所示。按

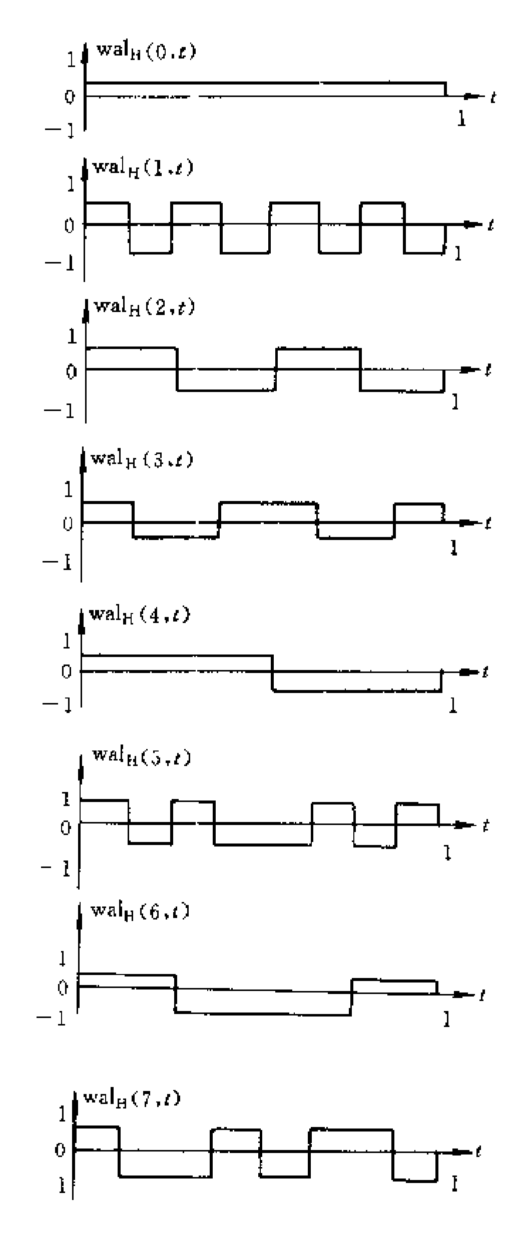


图 3-13　按哈达玛排列的沃尔什函数

哈达玛排列的沃尔什函数也可以写成矩阵式：

$$H_H(3)=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&1&1&1&-1&-1&-1&-1\\1&-1&1&-1&-1&1&-1&1\\1&1&-1&-1&-1&-1&1&1\\1&-1&-1&1&-1&1&1&-1\end{bmatrix} \tag{3-117}$$

按哈达玛排列的沃尔什函数有如下一些特点：

1）从 2 阶哈达玛矩阵可得到 2 个沃尔什函数，从 4 阶哈达玛矩阵可得到 4 个沃尔什函数，一般地说，2^n 阶哈达玛矩阵可得到 2^n 个沃尔什函数。

2）由不同阶数的哈达玛矩阵得到的沃尔什函数排列顺序是不同的。例如，从 $H_H(4)$ 得到的沃尔什函数 $\mathrm{wal}_H(2,t)$ 并不是从 $H_H(8)$ 得到的 $\mathrm{wal}_H(2,t)$，而是从 $H_H(8)$ 得到的 $\mathrm{wal}_H(4,t)$。

3）由于哈达玛矩阵的简单的递推关系，使得按哈达玛排列的沃尔什函数特别容易记忆。

按哈达玛排列的沃尔什函数也可以由拉德梅克函数产生，解析式如式(3-118)所示：

$$\mathrm{wal}_H(i,t)=\prod_{k=0}^{p-1}[R(k+1,t)]^{\langle i_k\rangle} \tag{3-118}$$

式中 $R(k+1,t)$ 仍然是拉德梅克函数，$\langle i_k\rangle$ 是把 i 的自然二进码反写后的第 k 位数字，并且 $i_k\in\{0,1\}$，也就是

$$(i)=(i_{n-1}i_{n-2}\cdots i_2i_1i_0)_B$$

反写后为

$$\langle i\rangle=(i_0i_1i_2\cdots\ i_{n-2}\ i_{n-1})$$

… 第1位 第0位

例　求 $p=3$ 时，$\mathrm{wal}_H(6,t)$ 的波形。

第一种方法可用比较简单的方法是写出 $2^n=2^3=8$ 阶哈达玛矩阵 $H_H(8)$，并自上而下从 0 数起至第 6 行就是 $\mathrm{wal}_H(6,t)$。

第二种方法是应用数学解析式。因为：

$$i=6=(110)_B$$

所以：

$$\langle i\rangle=(0\quad 1\quad 1)$$

$$\langle i_2\rangle\quad\langle i_1\rangle\quad\langle i_0\rangle$$

代入式(3-118)得：

$$\mathrm{wal}_H(6,t)=[R(1,t)]^{\langle i_0\rangle}\cdot[R(2,t)]^{\langle i_1\rangle}\cdot[R(3,t)]^{\langle i_2\rangle}=R(1,t)\cdot R(2,t)$$

三种定义下的沃尔什函数，尽管它们的排列顺序各不相同，但三种排序方法得到的沃尔什函数是有一定关系的。它们之间的关系如表 3-2 和图 3-14 所示。

表 3-2　三种排列的前 8 个沃尔什函数之间的关系表

$\text{wal}_H(i,t)$	$\text{wal}_W(i,t)$	$\text{wal}_P(i,t)$
$\text{wal}_H(0,t)$	$\text{wal}_W(0,t)$	$\text{wal}_P(0,t)$
$\text{wal}_H(1,t)$	$\text{wal}_W(7,t)$	$\text{wal}_P(4,t)$
$\text{wal}_H(2,t)$	$\text{wal}_W(3,t)$	$\text{wal}_P(2,t)$
$\text{wal}_H(3,t)$	$\text{wal}_W(4,t)$	$\text{wal}_P(6,t)$
$\text{wal}_H(4,t)$	$\text{wal}_W(1,t)$	$\text{wal}_P(1,t)$
$\text{wal}_H(5,t)$	$\text{wal}_W(6,t)$	$\text{wal}_P(5,t)$
$\text{wal}_H(6,t)$	$\text{wal}_W(2,t)$	$\text{wal}_P(3,t)$
$\text{wal}_H(7,t)$	$\text{wal}_W(5,t)$	$\text{wal}_P(7,t)$

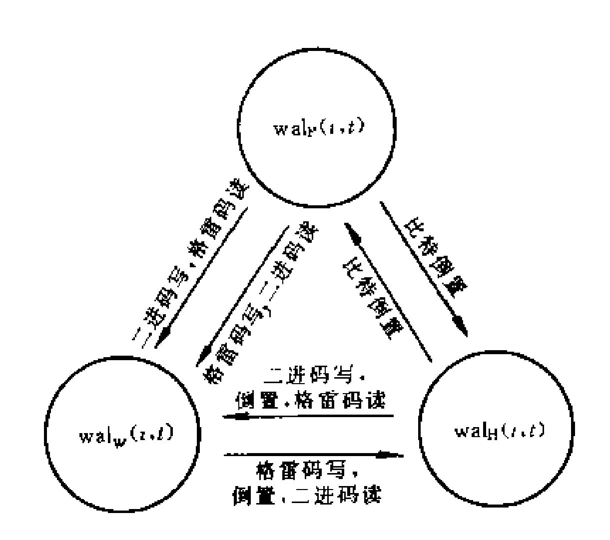


图 3-14　三种定义的沃尔什函数序号间的关系

例如，$\text{wal}_P(2,t)$ 的 $\text{wal}_W(i,t)$ 和 $\text{wal}_H(i,t)$ 间的关系如下：$2=(010)_B$，按格雷码读，即 $(010)_G=3$，所以 $\text{wal}_P(2,t)$ 就是 $\text{wal}_W(3,t)$，(010) 比特倒置后为 (010)，按二进码读仍为 2，则 $\text{wal}_P(2,t)$ 就是 $\text{wal}_H(2,t)$，其他以此类推。以上就是沃尔什函数三种定义之间的关系。

3.3.4 沃尔什函数的性质

沃尔什函数有下面一些主要性质。

1) 在区间[0,1]内有下式成立：

$$\int_0^1 \text{wal}(0,t)\text{d}t = 1 \tag{3-119}$$

$$\int_0^1 \text{wal}(i,t)\text{d}t = 0 \qquad i = 1,2,\cdots \tag{3-120}$$

$$[\text{wal}(i,t)]^2 = 1 \qquad i = 0,1,2,\cdots \tag{3-121}$$

这说明在[0,1]区间内除了 $\text{wal}(0,t)$ 外，其他沃尔什函数取 +1 和 −1 的时间是相等的。

2) 在区间[0,1]的第一小段时间内（通常称为时隙）沃尔什函数总是取 +1。

3) 沃尔什函数有如下乘法定理，

$$\text{wal}(i,t)\cdot\text{wal}(j,t) = \text{wal}(i\oplus j,t) \tag{3-122}$$

并且，该定理服从结合律，

$$[\mathrm{wal}(i,t)\cdot\mathrm{wal}(j,t)]\cdot\mathrm{wal}(k,t)=\mathrm{wal}(i,t)\cdot[\mathrm{wal}(j,t)\cdot\mathrm{wal}(k,t)]$$

$$i,j,k=0,1,2\cdots,2^p-1 \tag{3-123}$$

证明 由定义式

$$\begin{aligned}\mathrm{wal}(i,t)\cdot\mathrm{wal}(j,t)&=\prod_{k=0}^{p-1}[R(k+1,t)]^{g(i)_k}\cdot\prod_{k=0}^{p-1}[R(k+1,t)]^{g(j)_k}\\&=\prod_{k=0}^{p-1}[R(k+1,t)]^{g(i)_k+g(j)_k}\end{aligned}$$

但是，

$$\begin{gathered}g(i)_k,\ g(j)_k\in\{0,1\}\\ [R(k+1,t)]^{1+1}=[R(k+1,t)]^2=1\\ [R(k+1,t)]^{1\oplus1}=[R(k+1,t)]^0=1\\ [R(k+1,t)]^{1+0}=[R(k+1,t)]^{1\oplus0}\\ [R(k+1,t)]^{0+1}=[R(k+1,t)]^{0\oplus1}\end{gathered}$$

因此，

$$\begin{aligned}\mathrm{wal}(i,t)\cdot\mathrm{wal}(j,t)&=\prod_{k=0}^{p-1}[R(k+1,t)]^{g(i)_k+g(j)_k}\\&=\prod_{k=0}^{p-1}[R(k+1,t)]^{g(i)_k\oplus g(j)_k}=\mathrm{wal}(i\oplus j,t)\end{aligned}$$

以上便是乘法定理的证明。

4）沃尔什函数有归一化正交性：

$$\int_0^1\mathrm{wal}(i,t)\cdot\mathrm{wal}(j,t)\mathrm{d}t=\begin{cases}0 & i\neq j\\ 1 & i=j\end{cases} \tag{3-124}$$

证明 由乘法定理有

$$\int_0^1\mathrm{wal}(i,t)\cdot\mathrm{wal}(j,t)\mathrm{d}t=\int_0^1\mathrm{wal}(i\oplus j,t)\mathrm{d}t=\int_0^1\mathrm{wal}(l,t)\mathrm{d}t$$

其中 $i\oplus j=l$。

由于

$$\int_0^1\mathrm{wal}(0,t)\mathrm{d}t=1$$

$$\int_0^1\mathrm{wal}(i,t)\mathrm{d}t=0 \qquad i=1,2,\cdots$$

所以，当 $l=0$，即 $i=j$ 时，则

$$\int_0^1\mathrm{wal}(i,t)\cdot\mathrm{wal}(j,t)\mathrm{d}t=1$$

而当 $l\neq0$，即 $i\neq j$ 时，则

$$\int_0^1\mathrm{wal}(i,t)\cdot\mathrm{wal}(j,t)\mathrm{d}t=0$$

正交性得证。

5）沃尔什函数形成群

① 由沃尔什函数全体所组成的集合对于乘法运算而言形成一个可交换群。

② 由 2^n 个沃尔什函数 $\{\mathrm{wal}(0,t),\mathrm{wal}(1,t),\cdots,\mathrm{wal}(2^n-1,t)\}$，其中 n 为正整数，所组成的

子集，也形成一个可交换群，而且是上述群的一个子群。因为集合{wal(i,t)}满足形成群的四个公理，也就是说它满足封闭性；乘法结合律；存在一个单位元素 e，对于集合中的每一个元素 a，都可在该集合中找到一个逆元素 a^{-1}，使得 $a\cdot a^{-1}=e$，而 wal(i,t)的逆元素就是其本身。除此之外，沃尔什函数群满足乘法交换律，所以是一个可交换群。

6)
$$\mathrm{wal}(2^n i,t)=\mathrm{wal}(i,2^n t) \tag{3-125}$$

这里 n 为整数。这个性质可证明如下：

令
$$i=\sum_{k=0}^{p-1} i_k\cdot 2^k \qquad t<2^{p-1}$$

则
$$2^n i=2^n\cdot\sum_{k=0}^{p-1} i_k\cdot 2^k=\sum_{k=0}^{p-1} i_k\cdot 2^{n+k}$$

由定义
$$\mathrm{wal_W}(i,t)=\prod_{k=0}^{p-1}\mathrm{sgn}[\cos t_k\cdot 2^k\pi t]$$

可得

$$\mathrm{wal_W}(2^n i,t)=\prod_{k=0}^{p-1}\mathrm{sgn}[\cos i_k\cdot 2^{n+k}\pi t]$$
$$=\prod_{k=0}^{p-1}\mathrm{sgn}[\cos i_k\cdot 2^k\pi\cdot(2^n t)]=\mathrm{wal_W}(i,2^n t)$$

所以

$$\mathrm{wal}(2^n i,t)=\mathrm{wal}(i,2^n t)$$

7）对称性

$$\mathrm{wal}(i,t)=\mathrm{wal}(t,i) \tag{3-126}$$

这一性质只适用于离散沃尔什函数。

3.3.5 沃尔什变换

离散沃尔什变换可由下二式表达：

$$W(i)=\frac{1}{N}\sum_{t=0}^{N-1} f(t)\cdot\mathrm{wal}(i,t) \tag{3-127}$$

$$f(t)=\sum_{i=0}^{N-1} W(i)\cdot\mathrm{wal}(i,t) \tag{3-128}$$

离散沃尔什变换解析式写成矩阵式可得到沃尔什变换矩阵式：

$$\begin{bmatrix} W(0)\\ W(1)\\ \vdots\\ W(N-1)\end{bmatrix}=\frac{1}{N}[\mathrm{wal}(N)]\begin{bmatrix} f(0)\\ f(1)\\ \vdots\\ f(N-1)\end{bmatrix} \tag{3-129}$$

$$\begin{bmatrix} f(0)\\ f(1)\\ \vdots\\ f(N-1)\end{bmatrix}=[\mathrm{wal}(N)]\begin{bmatrix} f(0)\\ f(1)\\ \vdots\\ f(N-1)\end{bmatrix} \tag{3-130}$$

式中 wal(N)代表 N 阶沃尔什矩阵。

另外，沃尔什函数可写成如下形式：

$$\mathrm{wal}(i,t) = (-1)^{\sum_{k=0}^{p-1} t_{p-1-k}(i_{k+1} \oplus i_k)} \tag{3-131}$$

式中，$t=(t_{p-1}t_{p-2}\cdots t_k\cdots t_2t_1t_0)_B$，$i=(i_{p-1}i_{p-2}\cdots i_k\cdots i_2i_1i_0)_B$，$N=2^p$。因此，可得到指数形式的沃尔什变换式：

$$W(i)=\frac{1}{N}\sum_{t=0}^{N-1} f(t)\cdot(-1)^{\sum_{k=0}^{p-1} t_{p-1-k}(i_{k-1}\oplus i_k)} \tag{3-132}$$

$$f(t)=\sum_{n=0}^{N-1} W(i)(-1)^{\sum_{k=0}^{p-1} t_{p-1-k}(i_{k+1}\oplus i_k)} \tag{3-133}$$

以上是离散沃尔什变换的三种定义，其中矩阵式最为简洁。

3.3.6 离散沃尔什-哈达玛变换

由沃尔什函数的定义可知，按哈达玛排列的沃尔什函数与按沃尔什排列的沃尔什函数相比较只是排列顺序不同，其本质并没有什么不同。但是哈达玛矩阵具有简单的递推关系，也就是高阶矩阵可用低阶矩阵的直积得到，这就使得沃尔什-哈达玛变换有许多方便之处。因此，用得较多的是沃尔什-哈达玛变换。

离散沃尔什-哈达玛变换的定义可直接由沃尔什变换得到，只要用按哈达玛排列的沃尔什函数去代替沃尔什排列的沃尔什函数，就可以得其矩阵式如下：

$$\begin{bmatrix} W(0) \\ W(1) \\ \vdots \\ W(N-1) \end{bmatrix} = \frac{1}{N}[H(N)] \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix} \tag{3-134}$$

式中，$[W(0),W(1),W(2),\cdots,W(N-1)]^T$ 是沃尔什-哈达玛变换系数序列，$[f(0),f(1),f(2),\cdots,f(N-1)]^T$ 是时间序列，$N=2^p$，p 为正整数。式(3-134)的逆变换式如下：

$$\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix} = [H(N)] \begin{bmatrix} W(0) \\ W(1) \\ \vdots \\ W(N-1) \end{bmatrix} \tag{3-135}$$

例　将时间序列[0,0,1,1,0,0,1,1]做沃尔什-哈达玛变换及反变换。

$$\begin{bmatrix} W(0) \\ W(1) \\ W(2) \\ W(3) \\ W(4) \\ W(5) \\ W(6) \\ W(7) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \\ -1/2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

反变换为

$$
\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1/2 \\ 0 \\ -1/2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}
$$

3.3.7 离散沃尔什变换的性质

离散沃尔什变换有许多性质。下面把主要性质列举于下。为叙述方便起见，用$\{f(t)\}$表示时间序列，用$\{W(n)\}$表示变换系数序列，以$\{f(t)\} \Leftrightarrow \{W(n)\}$表示沃尔什变换对应关系。

(1) 线性

如果

$$\{f_1(t)\} \Leftrightarrow \{W_1(n)\}$$

$$\{f_2(t)\} \Leftrightarrow \{W_2(n)\}$$

则

$$a_1\{f_1(t)\} + a_2\{f_2(t)\} \Leftrightarrow a_1\{W_1(n)\} + a_2\{W_2(n)\} \tag{3-136}$$

其中 a_1、a_2 为常数。

(2) 模 2 移位性质

将时间序列$\{f(t)\}$作 l 位模 2 移位所得到的序列，我们称为模 2 移位序列。模 2 移位是这样实现的：

设：$\{f(t)\}=\{f(0),f(1),f(2),\cdots,f(N-1)\}$是周期长度为 N 的序列。作一个新的序列：

$$\{z(m)\}_l = \{z(0),z(1),z(2),\cdots,z(N-1)\} \tag{3-137}$$

其中 $z(m)=f(t\oplus l)$。此时，称$\{z(m)\}_l$ 是序列的位模 2 移位序列。

例 $\{f(t)\}=\{f(0),f(1),f(2),f(3),f(4),f(5),f(6),f(7)\}$

$N=8$，则：

$$\{z(m)\}_l=\{f(0\oplus 2),f(1\oplus 2),f(2\oplus 2),f(3\oplus 2),f(4\oplus 2),$$
$$f(5\oplus 2),f(6\oplus 2),f(7\oplus 2)\}$$

由于

$$0\oplus 2=(0\ 0\ 0)\oplus(0\ 1\ 0)=(0\ 1\ 0)_{二进}=(2)_{十进}$$
$$1\oplus 2=(0\ 0\ 1)\oplus(0\ 1\ 0)=(0\ 1\ 1)_{二进}=(3)_{十进}$$
$$2\oplus 2=(0\ 1\ 0)\oplus(0\ 1\ 0)=(0\ 0\ 0)_{二进}=(0)_{十进}$$
$$3\oplus 2=(0\ 1\ 1)\oplus(0\ 1\ 0)=(0\ 0\ 1)_{二进}=(1)_{十进}$$
$$4\oplus 2=(1\ 0\ 0)\oplus(0\ 1\ 0)=(1\ 1\ 0)_{二进}=(6)_{十进}$$
$$5\oplus 2=(1\ 0\ 1)\oplus(0\ 1\ 0)=(1\ 1\ 1)_{二进}=(7)_{十进}$$
$$6\oplus 2=(1\ 1\ 0)\oplus(0\ 1\ 0)=(1\ 0\ 0)_{二进}=(4)_{十进}$$
$$7\oplus 2=(1\ 1\ 1)\oplus(0\ 1\ 0)=(1\ 0\ 1)_{二进}=(5)_{十进}$$

所以

$$\{z(m)\}_2 = \{f(2), f(3), f(0), f(1), f(6), f(7), f(4), f(5)\}$$

同理，

$$\{z(m)\}_3 = \{f(3), f(2), f(1), f(0), f(7), f(6), f(5), f(4)\}$$

用矩阵表示为

$$[z]_1 = [M_1][f] \tag{3-138}$$

式中 $[f] = \{f(t)\}^{\mathrm{T}} \qquad [z]_1 = \{z(m)\}_1^{\mathrm{T}}$

$$[M_1] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3-139}$$

$$[z]_2 = [M_2][f] \tag{3-140}$$

$$[M_2] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3-141}$$

按照模 2 和的性质，可知

$$[M]^{\mathrm{T}}[M] = [I]$$

这里$[I]$是么阵。

模 2 移位性质是指下面的关系，如果$\{f(t)\} \Leftrightarrow \{W(n)\}$，并且$\{z(t)\}_l$是 $f(t)$的模 2 移位序列，则

$$\{z(t)\}_l \Leftrightarrow \{W_l(n)\} \tag{3-142}$$

式中：$W_l(n) = \mathrm{wal}(n,l) \cdot W(n)$，$\mathrm{wal}(n,l)$是矩阵$[\mathrm{wal}]_{2^p}$中的第 n 行第 l 列的元素；$n=0,1,2,\cdots,N-1$；$t=0,1,2,\cdots,N-1$；$N=2^p$，p 是正整数。

此定理可证明如下：

令 $z(t)$为$[z(t)]_l$的元素，$[z(t)]_l$是$[f(t)]$的模 2 移位序列，则

$$W_l(n) = \frac{1}{N}\sum_{t=0}^{N-1} z(t) \cdot \mathrm{wal}(n,t)$$

$$= \frac{1}{N}\sum_{t=0}^{N-1} f(t \oplus l) \cdot \mathrm{wal}(n,t)$$

令 $r=t\ominus l$，则有 $t=r\oplus l$，并且当 t 取值由 0 到 $N-1$ 时，r 也取同样的值，只是取值的顺序不同而已。于是可写成如下形式：

$$
\begin{aligned}
W_z(n) &= \frac{1}{N}\sum_{r=0}^{N-1} f(r)\cdot \mathrm{wal}(n, r\oplus l) \\
&= \frac{1}{N}\sum_{r=0}^{N-1} f(r)\cdot \mathrm{wal}(n,r)\cdot \mathrm{wal}(n,l) \\
&= \mathrm{wal}(n,l)\left[\frac{1}{N}\sum_{r=0}^{N-1} f(r)\cdot \mathrm{wal}(n,r)\right] \\
&= \mathrm{wal}(n,l)\left[\frac{1}{N}\sum_{t=0}^{N-1} f(t)\cdot \mathrm{wal}(n,t)\right] \\
&= \mathrm{wal}(n,l)\cdot W(n)
\end{aligned}
$$

所以，证明$\{z(t)\}_l \Leftrightarrow \{W_z(n)\}$。又因为$[W_z(n)]^2=[\mathrm{wal}(n,l)]^2\cdot[W(n)]^2=[W(n)]^2$，说明$[W_z(n)]^2$与$l$无关。也就是说，模2移位后的序列，作沃尔什变换后，所得到的第n个系数的平方$[W_z(n)]^2$与模2移位的移位位数无关。$[W_z(n)]^2$仍然等于$[W(n)]^2$。因此，模2移位定理（或称为并元移位定理）又可表达为输入序列$\{f(t)\}$模2移位后的功率谱是不变的。

例如，设输入序列$\{f(t)\}=\{0,0,1,1,0,0,1,1\}$，对此序列作$l=3$的模2移位，得

$$
\{z(t)\}_3 = \{f(t\oplus 3)\} = \{1,1,0,0,1,1,0,0\}
$$

作沃尔什变换得

$$
\{W(n)\} = \left\{\frac{1}{2},0,0,-\frac{1}{2},0,0,0,0\right\}
$$

根据$W_z(n)=\mathrm{wal}(n,l)\cdot W(n)$可得

$$
\begin{aligned}
\{W_z(n)\} &= \{\mathrm{wal}(n,l)\cdot W(n)\} \\
&= \{\mathrm{wal}(n,3)\cdot W(n)\} \\
&= \left\{\frac{1}{2},0,0,\frac{1}{2},0,0,0,0\right\}
\end{aligned}
$$

从上面结果可知$[W(0)]^2=\left(\frac{1}{2}\right)^2=\frac{1}{4}$，$[W(3)]^2=\left(-\frac{1}{2}\right)^2=\frac{1}{4}$，而$[W_z(0)]^2=\left(\frac{1}{2}\right)^2=\frac{1}{4}$，$[W_z(3)]^2=\left(\frac{1}{2}\right)^2=\frac{1}{4}$。可见$n$相同时，功率也相同，也就是说功率列率谱是不变的。

(3) 模2卷积定理(时间)

在讨论下面的定理之前，首先说明一下模2移位卷积与模2移位相关的概念。

令$\{f_1(t)\}$和$\{f_2(t)\}$是两个长度相同的周期性序列。用下面两式来定义两个序列的模2移位卷积和模2移位相关：

$$
\begin{aligned}
C_{12}(t) &= \frac{1}{N}\sum_{l=0}^{N-1} f_1(l)\cdot f_2(t\ominus l) \\
&= \frac{1}{N}\sum_{l=0}^{N-1} f_1(l)\cdot f_2(t\oplus l)
\end{aligned} \tag{3-143}
$$

式中$C_{12}(t)$为模2卷积的代表符号，$\ominus$为模2减运算符，它的运算结果与模2加一样。模2移位相关的定义式如式(3-144)所示：

$$
K_{12}(t) = \frac{1}{N}\sum_{l=0}^{N-1} f_1(l)\cdot f_2(t\oplus l) \tag{3-144}
$$

其中$K_{12}(t)$表示模2移位相关，$f_2(t\oplus l)$是$f_2(t)$的模2移位序列。

由式(3-143)和(3-144)可见，模2移位卷积和模2移位相关具有相同的结果，即：

$$K_{12}(t)=C_{12}(t)=\frac{1}{N}\sum_{l=0}^{N-1}f_1(l)\cdot f_2(t\oplus l)$$

下面讨论模 2 移位卷积定理。

如果

$$\{f_1(t)\}\Leftrightarrow\{W_1(n)\}$$
$$\{f_2(t)\}\Leftrightarrow\{W_2(n)\}$$

则

$$\{C_{12}(t)\}\Leftrightarrow\{W_1(n)\cdot W_2(n)\} \tag{3-145}$$

如果用 $\mathscr{W}$ 代表作沃尔什变换，则：

$$\begin{aligned}
\mathscr{W}[C_{12}(t)]&=\frac{1}{N}\sum_{t=0}^{N-1}C_{12}(t)\cdot \mathrm{wal}(n,t)\\
&=\frac{1}{N}\sum_{t=0}^{N-1}\left[\frac{1}{N}\sum_{t=0}^{N-1}f_1(l)\cdot f_2(t\oplus l)\right]\cdot \mathrm{wal}(n,t)\\
&=\frac{1}{N}\sum_{t=0}^{N-1}f_1(l)\left[\frac{1}{N}\sum_{t=0}^{N-1}f_2(t\oplus l)\cdot \mathrm{wal}(n,t)\right]\\
&=\frac{1}{N}\sum_{t=0}^{N-1}f_1(l)\cdot W_2(n)\cdot \mathrm{wal}(n,l)\\
&=W_2(n)\cdot\left[\frac{1}{N}\sum_{t=0}^{N-1}f_1(l)\cdot \mathrm{wal}(n,t)\right]=W_2(n)\cdot W_1(n)
\end{aligned}$$

所以，证明

$$\{C_{12}(t)\}\Leftrightarrow\{W_1(n)\cdot W_2(n)\}$$

(4) 模 2 移位列率卷积定理

模 2 移位列率卷积由下式来表示：

$$W_1(n)*W_2(n)=\sum_{r=0}^{N-1}W_1(r)\cdot W_2(r\oplus n) \tag{3-146}$$

依照模 2 时间卷积定理，模 2 移位列率卷积定理如下：

如果

$$\{f_1(t)\}\Leftrightarrow\{W_1(n)\}$$
$$\{f_2(t)\}\Leftrightarrow\{W_2(n)\}$$

则

$$\{W_1(n)*W_2(n)\}\Leftrightarrow\{f_1(t)\cdot f_2(t)\} \tag{3-147}$$

这个仿照模 2 移位时间卷积定理的证明方法不难得到证明。

(5) 模 2 移位自相关定理

从模 2 移位时间卷积(相关)定理可以得到模 2 移位自相关定理。只要把定理中的$\{f_2(t)\}$和$\{W_2(n)\}$换成$\{f_1(t)\}$和$\{W_1(n)\}$便立即可以得到模 2 移位自相关定理：

$$\{K_{11}(t)\}\Leftrightarrow\{W_1^2(n)\} \tag{3-148}$$

其证明方法也与模 2 移位时间卷积定理的证明方法一样。

从式(3-148)可以建立一个重要概念：模 2 移位自相关序列的沃尔什变换等于序列的功率谱。也就是说，模 2 移位下的自相关序列的沃尔什变换正好与序列的功率谱相符合。与傅里叶变换相比较，模 2 移位下的自相关与沃尔什谱的关系相当于线性移位下的自相关序列的离散

傅里叶变换与其功率谱的关系。

(6) 帕斯维尔定理

如果

$$\{f_1(t)\} \Leftrightarrow \{W_1(n)\}$$

则

$$\frac{1}{N}\sum_{t=0}^{N-1} f_1^2(t) = \sum_{n=0}^{N-1} W_1^2(n) \tag{3-149}$$

证明 设 $\{K_{11}(t)\} \Leftrightarrow \{W(n)\}$

则

$$K_{11}(t) = \sum_{t=0}^{N-1} W(n) \cdot \mathrm{wal}(n,t)$$

$$= \sum_{t=0}^{N-1} W_1^2(n) \cdot \mathrm{wal}(n,t)$$

因为 $K_{11}(t)$是自相关函数，所以

$$\{K_{11}(t)\} \Leftrightarrow \{W_1^2(n)\}$$

又由于

$$K_{11}(t) = \frac{1}{N}\sum_{l=0}^{N-1} f_1(l) \cdot f_1(t \oplus l)$$

所以

$$\frac{1}{N}\sum_{l=0}^{N-1} f_1(l) \cdot f_1(t \ominus l) = \frac{1}{N}\sum_{t=0}^{N-1} W_1^2(n) \cdot \mathrm{wal}(n,t)$$

如果令 $t=0$，则

$$\frac{1}{N}\sum_{l=0}^{N-1} f_1^2(l) = \sum_{n=0}^{N-1} W_1^2(n)$$

由于 l 仅是求和运算的变量，因此将 l 换成 t，即可得：

$$\frac{1}{N}\sum_{t=0}^{N-1} f_1^2(t) = \sum_{n=0}^{N-1} W_1^2(n)$$

(7) 循环移位定理

把序列$\{f(t)\}$循环地向左移若干位，例如移 l 位，$l=1,2,\cdots,N-1$，这样得到的序列叫循环移位序列。如果用$\{z(t)_l\}$来表示循环移位序列，则

$$\{z(t)_l\} = \{f(l), f(l+1), \cdots, f(l-2), f(l-1)\} \tag{3-150}$$

$$l = 1,2,\cdots,N-1$$

例如，有一个 $N=8$ 的序列$\{f(t)\}=\{f(0), f(1), f(2), f(3), f(4), f(5), f(6), f(7)\}$，当 $l=5$，$l=3$ 的循环移位序列分别为

$$\{z(t)_5\} = \{f(5), f(6), f(7), f(0), f(1), f(2), f(3), f(4)\}$$

$$\{z(t)_3\} = \{f(3), f(4), f(5), f(6), f(7), f(0), f(1), f(2)\}$$

循环移位定理的内容如下：

如果$\{f(t)\}$和它的循环移位序列$\{z(t)_l\}$的沃尔什-哈达玛变换分别是 $W_f(n)$和 $W_z(n)$，则

$$\left.\begin{aligned} &W_z^2(0) = W_f^2(0) \\ &\sum_{n=2^{(r-1)}}^{2^r-1} W_z^2(n) - \sum_{n=2^{(r-1)}}^{2^r-1} W_f^2(n) \end{aligned}\right\} \tag{3-151}$$

式中 $r=1,2,\cdots,p$，$p=\log_2 N$，$l=1,2,\cdots,N-1$。这个定理把序列$\{f(t)\}$的沃尔什-哈达玛变换系数 $W_f(n)$与循环移位序列$\{z(t)_l\}$的沃尔什-哈达玛变换系数 $W_z(n)$联系了起来。即某些$W_f^2(n)$之和与 $W_z^2(n)$之和是相等的。所以这个定理又称为沃尔什-哈达玛变换的循环移位不变性。下面用一个例子来说明本定理的意义。

例如，设$\{f(t)\}=\{0,0,1,1,0,0,1,1\}$，经沃尔什-哈达玛变换后的系数序列为

$$\{W_f(n)\} = \left\{\frac{1}{2},0,-\frac{1}{2},0,0,0,0,0\right\}$$

现将$\{f(t)\}$做 $l=3$ 的循环移位，则

$$\{z(t)_3\} = \{1,0,0,1,1,0,0,1\}$$

经沃尔什-哈达玛变换后的系数序列为

$$\{W_z(n)\} = \left\{\frac{1}{2},0,0,\frac{1}{2},0,0,0,0\right\}$$

从两个序列 $W_f(n)$与 $W_z(n)$可以看出 $W_z(0)=\dfrac{1}{2}$，$W_f(0)=\dfrac{1}{2}$，所以

$$W_z^2(0) = W_f^2(0) = \frac{1}{4}$$

当 $r=1$ 时，则

$$\sum_{n=2^{(r-1)}}^{2^r-1} W_z^2(n) = W_z^2(1) = 0$$

$$\sum_{n=2^{(r-1)}}^{2^r-1} W_f^2(n) = W_f^2(1) = 0$$

所以，

$$W_z^2(1) = W_f^2(1) = 0$$

当 $r=2$ 时，则

$$\sum_{n=2^{(r-1)}}^{2^r-1} W_z^2(n) = W_z^2(2) + W_z^2(3) = \frac{1}{4}$$

$$\sum_{n=2^{(r-1)}}^{2^r-1} W_f^2(n) = W_f^2(2) + W_f^2(3) = \frac{1}{4}$$

所以，

$$W_z^2(2) + W_z^2(3) = W_f^2(2) + W_f^2(3) = \frac{1}{4}$$

当 $r=3$ 时，则

$$\sum_{n=2^{(r-1)}}^{2^r-1} W_z^2(n) = W_z^2(4) + W_z^2(5) + W_z^2(6) + W_z^2(7) = 0$$

$$\sum_{n=2^{(r-1)}}^{2^r-1} W_f^2(n) = W_f^2(4) + W_f^2(5) + W_f^2(6) + W_f^2(7) = 0$$

所以，

$$W_z^2(4) + W_z^2(5) + W_z^2(6) + W_z^2(7) = W_f^2(4) + W_f^2(5) + W_f^2(6) + W_f^2(7)$$

显然，这些关系符合循环移位定理。

需要特别指出的是，这个定理只适用于沃尔什-哈达玛变换。此定理的更加一般性的证明，请参阅有关书籍，在此不再赘述。

3.3.8 快速沃尔什变换

离散傅里叶变换有快速算法。同样，离散沃尔什变换也有快速算法。利用快速算法，完成一次变换只须 $N\log_2 N$ 次加减法，运算速度可大大提高。当然，快速算法只是一种运算方法，就变换本身来说快速变换与非快速变换是没有区别的。由于沃尔什-哈达玛变换有清晰的分解过程，而且快速沃尔什变换可由沃尔什-哈达玛变换修改得到，所以下面着重讨论沃尔什-哈达玛快速变换。

由离散沃尔什-哈达玛变换的定义可知：

$$[W_H(n)] = \frac{1}{N}[H][f(t)] \tag{3-152}$$

式中 $N=2^p$，p 为正整数。

这里以 8 阶沃尔什-哈达玛变换为例，讨论其分解过程及快速算法。由克罗内克积可知：

$$\begin{aligned}
[H_8] &= [H_2] \otimes [H_4] \\
&= \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} \\
&= \begin{bmatrix} H_4 & 0 \\ 0 & H_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_4 & \boldsymbol{I}_4 \\ \boldsymbol{I}_4 & -\boldsymbol{I}_4 \end{bmatrix} \\
&= \begin{bmatrix} H_2 & H_2 & 0 & 0 \\ H_2 & -H_2 & 0 & 0 \\ 0 & 0 & H_2 & H_2 \\ 0 & 0 & H_2 & -H_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_4 & \boldsymbol{I}_4 \\ \boldsymbol{I}_4 & -\boldsymbol{I}_4 \end{bmatrix} \\
&= \begin{bmatrix} H_2 & 0 & 0 & 0 \\ 0 & H_2 & 0 & 0 \\ 0 & 0 & H_2 & 0 \\ 0 & 0 & 0 & H_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{I}_2 & 0 & 0 \\ \boldsymbol{I}_2 & -\boldsymbol{I}_2 & 0 & 0 \\ 0 & 0 & \boldsymbol{I}_2 & \boldsymbol{I}_2 \\ 0 & 0 & \boldsymbol{I}_2 & -\boldsymbol{I}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_4 & \boldsymbol{I}_4 \\ \boldsymbol{I}_4 & -\boldsymbol{I}_4 \end{bmatrix} \\
&= [G_0][G_1][G_2]
\end{aligned} \tag{3-153}$$

其中，

$$[G_0] = \begin{bmatrix} H_2 & 0 & 0 & 0 \\ 0 & H_2 & 0 & 0 \\ 0 & 0 & H_2 & 0 \\ 0 & 0 & 0 & H_2 \end{bmatrix} \tag{3-154}$$

$$[G_1] = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{I}_2 & 0 & 0 \\ \boldsymbol{I}_2 & -\boldsymbol{I}_2 & 0 & 0 \\ 0 & 0 & \boldsymbol{I}_2 & \boldsymbol{I}_2 \\ 0 & 0 & \boldsymbol{I}_2 & -\boldsymbol{I}_2 \end{bmatrix} \tag{3-155}$$

$$[G_2] = \begin{bmatrix} \boldsymbol{I}_4 & \boldsymbol{I}_4 \\ \boldsymbol{I}_4 & -\boldsymbol{I}_4 \end{bmatrix} \tag{3-156}$$

其中 $\boldsymbol{I}_2$、$\boldsymbol{I}_4$ 均为么阵。

由上面的分解有：

$$[W_H(n)] = \frac{1}{8}[G_0][G_1][G_2][f(t)] \tag{3-157}$$

令

$$[f_1(t)] = [G_2][f(t)]$$
$$[f_2(t)] = [G_1][f_1(t)]$$
$$[f_3(t)] = [G_0][f_2(t)]$$

则

$$[W_H(n)] = \frac{1}{8}[f_3(t)]$$

下面是具体计算$[f_1(t)]$,$[f_2(t)]$,$[f_3(t)]$的公式及流程图。

$$[f_1(t)] = [G_2][f(t)]$$

$$\begin{bmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \\ f_1(4) \\ f_1(5) \\ f_1(6) \\ f_1(7) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix} = \begin{bmatrix} f(0)+f(4) \\ f(1)+f(5) \\ f(2)+f(6) \\ f(3)+f(7) \\ f(0)-f(4) \\ f(1)-f(5) \\ f(2)-f(6) \\ f(3)-f(7) \end{bmatrix} \tag{3-158}$$

$$[f_2(t)] = [G_1][f_1(t)]$$

$$\begin{bmatrix} f_2(0) \\ f_2(1) \\ f_2(2) \\ f_2(3) \\ f_2(4) \\ f_2(5) \\ f_2(6) \\ f_2(7) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \\ f_1(4) \\ f_1(5) \\ f_1(6) \\ f_1(7) \end{bmatrix} = \begin{bmatrix} f_1(0)+f_1(2) \\ f_1(1)+f_1(3) \\ f_1(0)-f_1(2) \\ f_1(1)-f_1(3) \\ f_1(4)+f_1(6) \\ f_1(5)+f_1(7) \\ f_1(4)-f_1(6) \\ f_1(5)-f_1(7) \end{bmatrix} \tag{3-159}$$

$$[f_3(t)] = [G_0][f_2(t)]$$

$$\begin{bmatrix} f_3(0) \\ f_3(1) \\ f_3(2) \\ f_3(3) \\ f_3(4) \\ f_3(5) \\ f_3(6) \\ f_3(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} f_2(0) \\ f_2(1) \\ f_2(2) \\ f_2(3) \\ f_2(4) \\ f_2(5) \\ f_2(6) \\ f_2(7) \end{bmatrix} = \begin{bmatrix} f_2(0)+f_2(1) \\ f_2(0)-f_2(1) \\ f_2(2)+f_2(3) \\ f_2(2)-f_2(3) \\ f_2(4)+f_2(5) \\ f_2(4)-f_2(5) \\ f_2(6)+f_2(7) \\ f_2(6)-f_2(7) \end{bmatrix} \tag{3-160}$$

因为 $$[H_8]=[G_0][G_1][G_2]$$

而$[H_8]$、$[G_0]$，$[G_1]$，$[G_2]$是对称矩阵，即：

$$[H_8]^{\mathrm{T}}=[H_8]$$

$$[G_0]^{\mathrm{T}}=[G_0]$$

$$[G_1]^{\mathrm{T}}=[G_1]$$

$$[G_2]^{\mathrm{T}}=[G_2]$$

所以，

$$\begin{aligned}[H_8]&=[H_8]^{\mathrm{T}}=\{[G_0][G_0][G_0]\}^{\mathrm{T}}\\&=[G_2]^{\mathrm{T}}[G_1]^{\mathrm{T}}[G_0]^{\mathrm{T}}\\&=[G_2][G_1][G_0]\end{aligned}$$

$$[W_{\mathrm{H}}(n)]=\frac{1}{8}[G_2][G_1][G_0][f(t)] \tag{3-161}$$

$$[f_1(t)]=[G_0][f(t)]$$

$$[f_2(t)]=[G_1][f_1(t)]$$

$$[f_3(t)]=[G_2][f_2(t)]$$

$$[W_{\mathrm{H}}(n)]=\frac{1}{8}[f_3(t)]$$

由此可得到另一种蝶形运算流程图，见图 3-15。

对于一般情况，$N=2^p$，$p=0,1,2,\cdots$，则矩阵$[H_{2^p}]$可分解成 p 个矩阵$[G_p]$之乘积，即：

$$\begin{aligned}[H_{2^p}]&=\prod_{p=0}^{p-1}[G_p]=[G_0][G_1][G_2]\cdots[G_{p-1}]\\&=[G_{p-1}][G_{p-2}]\cdots[G_1][G_0]\end{aligned} \tag{3-162}$$

所以，任意 2^p 阶快速沃尔什-哈达玛变换蝶式流图不难用上述方法引伸。

3.3.9 多维变换

在图像处理中广泛运用的是二维变换，因此，下面对二维沃尔什-哈达玛变换作一介绍。二维沃尔什-哈达玛变换的指数式如下：

$$W_{xy}(u,v)=\frac{1}{N_x}\frac{1}{N_y}\sum_{y=0}^{N_y-1}\sum_{x=0}^{N_x-1}f(x,y)\cdot(-1)^{\sum\limits_{r=0}^{p_x-1}x_ru_r+\sum\limits_{s=0}^{p_y-1}y_sv_s} \tag{3-163}$$

式中：$f(x,y)$代表图像的像素，x、y 是该像素在空间中的位置坐标；$W_{xy}(u,v)$代表变换系数；x、$u=0,1,2,\cdots,N_x-1$；$N_x=2^{p_x}$，p_x 为正整数；x_r，u_r 是 x、u 的二进码的第 r 位数字，$\{x_r,u_r\}\in\{0,1\}$；y，$v=0,1,2,\cdots,N_y-1$，$N_y=2^{p_y}$，p_y 为正整数；y_s、v_s 为 y，v 的二进码的第 s 位数字$\{y_s,v_s\}\in\{0,1\}$。

二维沃尔什-哈达玛变换的逆变换式为

$$f(x,y)=\sum_{v=0}^{N_y-1}\sum_{u=0}^{N_x-1}W_{xy}(u,v)(-1)^{\sum\limits_{r=0}^{p_x-1}x_ru_r+\sum\limits_{s=0}^{p_y-1}y_sv_s} \tag{3-164}$$

式中各参数的意义与正变式相同。

二维输入数据可写成矩阵形式：

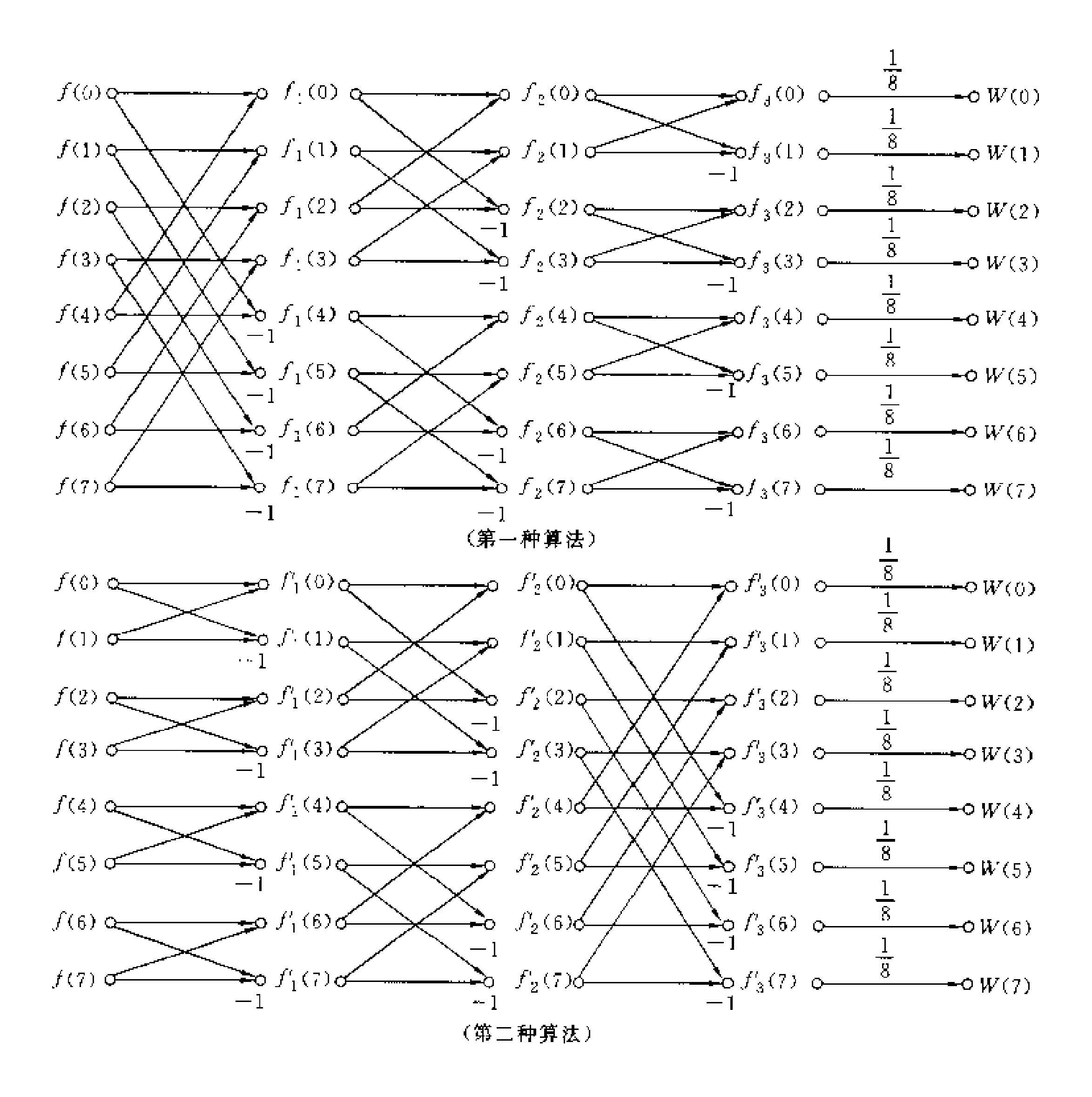


图 3-15　快速沃尔什-哈达玛变换信号流图(二种算法)

$$[f(x,y)] = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N_y-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N_y-1) \\ \cdots & \cdots & \cdots & \cdots \\ f(N_x-1,0) & f(N_x-1,1) & \cdots & f(N_x-1,N_y-1) \end{bmatrix} \tag{3-165}$$

首先,按公式(3-163)求第一个求和结果,即

$$\begin{aligned} \frac{1}{N_x}\sum_{x=0}^{N_x-1} f(x,y)(-1)^{\sum_{i=0}^{p_x-1} x_i u_i} &= \frac{1}{N_x}\sum_{x=0}^{N_x-1} f(x,y)(-1)^{(x,u)} \\ &= \frac{1}{N_x}\{f(0,y)(-1)^{(0,u)} + f(1,y)(-1)^{(1,u)} \\ &\quad + \cdots + f(N_x-1,y)(-1)^{(N_x-1,u)}\} \end{aligned} \tag{3-166}$$

式(3-166)的右边显然是某一列的沃尔什-哈达玛变换。如果 $y=0$ 则为输入阵列的第一列沃尔什-哈达玛变换，$y=1$ 则为第二列等等。如果令

$$W_x(u,y) = \frac{1}{N_x}\sum_{x=0}^{N_x-1} f(x,y)(-1)^{(x,u)} \tag{3-167}$$

那么,第一个和式所得到的变换系数可写成一个矩阵形式：

$$[W_x(u,y)] = \begin{bmatrix} W_x(0,0) & W_x(0,1) & \cdots & W_x(0,N_y-1) \\ W_x(1,0) & W_x(1,1) & \cdots & W_x(1,N_y-1) \\ \cdots & \cdots & \cdots & \cdots \\ W_x(N_x-1,0) & W_x(N_x-1,1) & \cdots & W_x(N_x-1,N_y-1) \end{bmatrix} \tag{3-168}$$

将上述数算完之后再看第二个求和，即

$$\begin{aligned} W_{xy}(u,v) &= \frac{1}{N_y}\sum_{y=0}^{N_y-1} W_x(u,y)(-1)^{\langle y,v\rangle} \\ &= \frac{1}{N_y}\{W_x(u,0)(-1)^{\langle 0,v\rangle} + W_x(u,1)(-1)^{\langle 1,v\rangle} \\ &\quad + \cdots + W_x(u,N_y-1)(-1)^{\langle N_y-1,v\rangle}\} \end{aligned} \tag{3-169}$$

式中 $\langle y,v\rangle = \sum_{i=0}^{p_y-1} y_i v_i$。

式(3-169)说明，$W_{xy}(u,v)$ 可以从计算 $[W_x(u,y)]$ 的每一行的沃尔什-哈达玛变换得到。其结果产生 N_xN_y 个系数。其矩阵表达式如式(3-170)所示：

$$[W_{xy}(u,v)] = \begin{bmatrix} W_{xy}(0,0) & W_{xy}(0,1) & \cdots & W_{xy}(0,N_y-1) \\ W_{xy}(1,0) & W_{xy}(1,1) & \cdots & W_{xy}(1,N_y-1) \\ \cdots & \cdots & \cdots & \cdots \\ W_{xy}(N_x-1,0) & W_{xy}(N_x-1,1) & \cdots & W_{xy}(N_x-1,N_y-1) \end{bmatrix} \tag{3-170}$$

由上面的分析可见，二维沃尔什-哈达玛变换可用一维沃尔什-哈达玛变换计算，步骤如下：

1）以 $N=N_x$，对 $[f(x,y)]$ 中 N_y 个列中的每一列做变换，得到 $[W_x(u,y)]$；

2）以 $N=N_y$，对 $[W_x(u,y)]$ 中 N_x 行的每一行作变换，即可得到二维变换系数 $[W_{xy}(u,v)]$。根据这一步骤，便可以利用一维快速沃尔什-哈达玛变换来完成二维沃尔什-哈达玛变换的计算。

另外一种计算方法是将二维沃尔什-哈达玛变换当做一维来计算。这种方法是将数据矩阵的各列依次顺序排列，这样就形成由 N_xN 个元素的列矩阵，然后再按照一维沃尔什-哈达玛变换方法来计算。下面用实例说明两种计算方法。

例　设数据矩阵如下：

$$[f(x,y)] = \begin{bmatrix} 1 & 1 & 3 & 1 \\ 2 & 1 & 2 & 2 \end{bmatrix}$$

求 $[f(x,y)]$ 的二维沃尔什-哈达玛变换。

首先对 $[f(x,y)]$ 的每一列作变换：

第一列

$$W_x(u,0) = \frac{1}{2}[H_2]\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

第二列

$$W_x(u,1) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

第三列

$$W_x(u,2) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 \\ 2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

第四列

$$W_x(u,3)=\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 2\end{bmatrix}=\frac{1}{2}\begin{bmatrix}3\\ -1\end{bmatrix}$$

所以

$$[W_x(u,y)]=\frac{1}{2}\begin{bmatrix}3 & 2 & 5 & 3\\ -1 & 0 & 1 & -1\end{bmatrix}$$

对$[W_x(u,y)]$每一行作变换：

第一行

$$W_x(u,0)=\frac{1}{4}\times\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}3\\ 2\\ 5\\ 3\end{bmatrix}=\frac{1}{8}\begin{bmatrix}13\\ 3\\ -3\\ -1\end{bmatrix}$$

第二行

$$W_x(u,1)=\frac{1}{4}\times\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}-1\\ 0\\ 1\\ -1\end{bmatrix}=\frac{1}{8}\begin{bmatrix}-1\\ 1\\ -1\\ -3\end{bmatrix}$$

最后得到二维变换系数矩阵：

$$[W_{xy}(u,v)]=\frac{1}{8}\begin{bmatrix}13 & 3 & -3 & -1\\ -1 & 1 & -1 & -3\end{bmatrix}$$

以上是采用第一种算法得到的结果。

第二种算法如下：

将$[f(x,y)]$改写成列矩阵$[Y]$，即

$$[Y]=\begin{bmatrix}1\\ 2\\ 1\\ 1\\ 3\\ 2\\ 1\\ 2\end{bmatrix}$$

对$[Y]$做一维变换：

$$\begin{bmatrix}[W_y(0)]\\ [W_y(1)]\\ [W_y(2)]\\ [W_y(3)]\\ [W_y(4)]\\ [W_y(5)]\\ [W_y(6)]\\ [W_y(7)]\end{bmatrix}=\frac{1}{8}\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 2\\ 1\\ 1\\ 3\\ 2\\ 1\\ 2\end{bmatrix}=\frac{1}{8}\begin{bmatrix}13\\ -1\\ 3\\ 1\\ -3\\ -1\\ -1\\ -3\end{bmatrix}$$

然后重排一下，

$$[W_{xy}(u,v)] = \frac{1}{8}\begin{bmatrix} 13 & 3 & -3 & -1 \\ -1 & 1 & -1 & -3 \end{bmatrix}$$

显然，与第一种算法得到的结果一致。

二维沃尔什-哈达玛变换的矩阵式定义如下：

$$[W_{xy}(u,v)] = \frac{1}{N_xN_y}[H_{2^{p_x}}][f(x,y)][H_{2^{p_y}}] \tag{3-171}$$

$$[f(x,y)] = [H_{2^{p_x}}][W_{xy}(u,v)][H_{2^{p_y}}] \tag{3-172}$$

式中$[H_{2^{p_x}}]$和$[H_{2^{p_y}}]$分别为 2^{p_x}阶和 2^{p_y}阶哈达玛矩阵。

附录 3 中给出一个二维快速沃尔什-哈达玛变换运算程序供参考。

3.4 哈尔函数及哈尔变换

在图像编码及数字滤波方面得到应用的另一种归一化正交函数是哈尔(Haar)函数。它的一个重要特点是收敛均匀而迅速。

3.4.1 哈尔函数的定义

哈尔函数是完备的、归一化的正交函数。在[0,1]区间内，har(0,t)为 1，har(1,t)在左半个区间内取值为 1，在右半个区间内取值为 −1。它的其他函数取 0 值和±1 乘以$\sqrt{2}$的幂，即取$\pm\sqrt{2}$，±2，$\pm 2\sqrt{2}$，±4 等。具体定义如下：

$$\mathrm{har}(0,t) = 1 \qquad 0 \leqslant t < 1$$

$$\mathrm{har}(1,t) = \begin{cases} 1 & 0 \leqslant t < \frac{1}{2} \\ -1 & \frac{1}{2} \leqslant t < 1 \end{cases}$$

$$\mathrm{har}(2,t) = \begin{cases} \sqrt{2} & 0 \leqslant t < \frac{1}{4} \\ -\sqrt{2} & \frac{1}{4} \leqslant t < \frac{1}{2} \\ 0 & \frac{1}{2} \leqslant t < 1 \end{cases}$$

$$\mathrm{har}(3,t) = \begin{cases} 0 & 0 \leqslant t < \frac{1}{2} \\ \sqrt{2} & \frac{1}{2} \leqslant t < \frac{3}{4} \\ -\sqrt{2} & \frac{3}{4} \leqslant t < 1 \end{cases}$$

一般情况，

$$\mathrm{har}(2^p+n,t) = \begin{cases} \sqrt{2^p} & \frac{n}{2^p} \leqslant t < \frac{(n+1/2)}{2^p} \\ -\sqrt{2^p}, & \frac{(n+1/2)}{2^p} \leqslant t < \frac{(n+1)}{2^p} \\ 0 & 其他 \end{cases} \tag{3-173}$$

$$p = 1,2,\cdots, \qquad n = 0,1,2,\cdots,2^p - 1$$

前 8 个哈尔函数的波形图示于图 3-16。

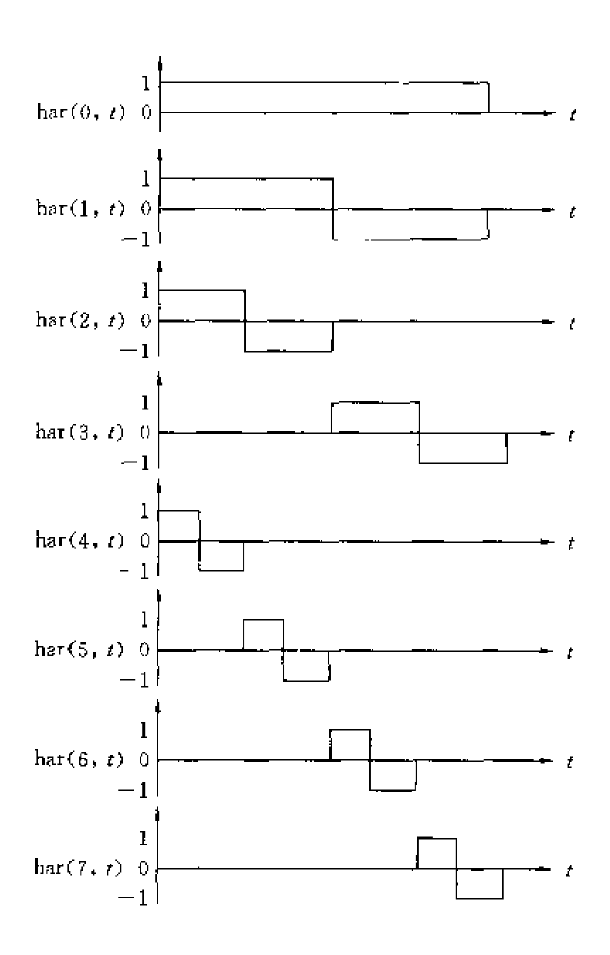


图 3-16 哈尔函数波形图

上述哈尔函数是定义在区间[0,1)内的，但是，我们可以以 1 为周期，将它延拓至整个时间轴上。

3.4.2 哈尔函数的性质

1. 归一化正交性

从图 3-16 我们可以看出哈尔函数的正交性。例如，哈尔函数 har(4,t)，har(5,t)，har(6,t)，har(7,t)在时间上是互不重迭的，因此，它们必然是正交的。另外，阶数不相同的两个哈尔函数，或者互不重迭(例如 har(3,t)和 har(4,t))；或者一个哈尔函数处于另一个哈尔函数的半周之内(例如 har(4,t)和 har(2,t))，这两种情况都是正交的。总的来说，哈尔函数的正交性可用下式表示：

$$\int_0^1 \text{har}(m,t)\text{har}(l,t)\text{d}t = \begin{cases} 1 & m = l \\ 0 & m \neq l \end{cases} \tag{3-174}$$

2. 哈尔级数

周期为 1 的连续函数 $f(t)$可以展开成哈尔级数，即

$$f(t) = \sum_{m=0}^{\infty} c(m)\text{har}(m,t) \tag{3-175}$$

其中

$$c(m) = \int_0^1 f(t)\mathrm{har}(m,t)\mathrm{d}t \tag{3-176}$$

哈尔级数的收敛性比沃尔什函数要好。

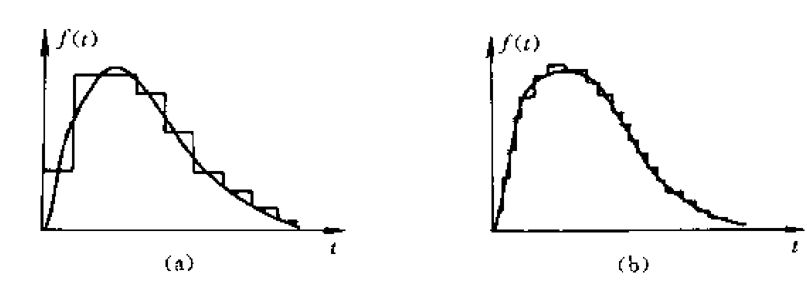


图 3-17 指数函数展开成有限哈尔函数

图 3-17 是用有限项哈尔级数去逼近指数函数的情况。从图中可见，逼近曲线是步长相等的阶梯波。步长为$\frac{1}{2^p}$，阶梯数目为 2^p，哈尔函数的最高阶为 p，由这个 p 确定了步长。从图可见，项数越多，逼近越好，这种增加项数的效果是很明显的。而在傅里叶级数和沃尔什级数中就没有这样直观、简单。

3. 帕斯维尔定理

因为哈尔函数是完备的正交函数，因此，帕斯维尔定理是成立的，即

$$\int_0^1 f^2(t)\mathrm{d}t = \sum_{m=0}^{+\infty} c^2(m) \tag{3-177}$$

4. 全域函数和区域函数

我们观察哈尔函数会发现，前两个哈尔函数 har(0,t)及 har(1,t)在整个正交区间内都有值，因此把它们称为全域函数。而其余的函数 har(2,t)，har(3,t)，har(4,t)等只在部分区间有值，因此，把它们称为区域函数。按上述定义来看，三角函数，沃尔什函数都是全域函数。从式(3-176)可以看出，全域函数 har(0,t)和 har(1,t)的系数 $c(0)$和 $c(1)$在整个正交区间内受 $f(t)$的影响；区域函数的系数 $c(2)$，$c(3)$等只受 $f(t)$部分值的影响。这样，如果用哈尔函数去逼近 $f(t)$，则全域函数在整个正交区间内起作用，而区域函数则在部分区域起作用。在工程应用中，如果我们希望将一个函数 $f(t)$的某一部分逼近得更好的话，那么，哈尔函数有独到之处。

3.4.3 哈尔变换及快速算法

把离散的哈尔函数写成矩阵形式就可得到哈尔矩阵。前 8 个哈尔函数组成的矩阵如式(3-178)所示：

$$[\mathrm{har}_8] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \tag{3-178}$$

哈尔正变换由下式定义：

$$\begin{bmatrix} H_a(0) \\ H_a(1) \\ \vdots \\ H_a(N-1) \end{bmatrix} = \frac{1}{N}[\mathrm{har}_{2^p}]\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix} \tag{3-179}$$

式中$[H_a(0),H_a(1),\cdots,H_a(N-1)]^T$是变换系数阵列，$[f(0),f(1),\cdots,f(N-1)]^T$是时间序列，$[\mathrm{har}_{2^p}]$是 2^p 阶哈尔矩阵，p 为正整数。

其逆变换如式(3-180)所示：

$$\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N-1) \end{bmatrix} = [\mathrm{har}_{2^p}]^{-1}\begin{bmatrix} H_a(0) \\ H_a(1) \\ H_a(2) \\ \vdots \\ H_a(N-1) \end{bmatrix} \tag{3-180}$$

式中$[\mathrm{har}_{2^p}]^{-1}$是$[\mathrm{har}_{2^p}]$的逆矩阵。由于哈尔矩阵不是对称矩阵，因此$[\mathrm{har}_{2^p}]^{-1}$不等于$[\mathrm{har}_{2^p}]$，所以哈尔正变换与逆变换是不相同的。

仿照沃尔什变换，利用矩阵因子分解之方法也可以得到快速哈尔变换。一般说来，快速哈尔变换的流程图并不是蝶形的，但是，我们可以用重新排序的方法构成哈尔变换的蝶式运算流程图。具体作法如下所述。

设原时间序列为$[f(0),f(1),f(2),f(3),f(4),f(5),f(6),f(7)]$，运算之前首先重新排序。令

$$[f'(t)] = [f'(0),f'(1),f'(2),f'(3),f'(4),f'(5),f'(6),f'(7)]$$

为排序后的新序列。具体排序方法如下：

1）将$[f(t)]$中元素的序号写成自然二进码；

2）将此二进码比特倒置；

3）把倒置后的二进码翻成十进制数字，这个数字就是新序列的序号。

例如，$f(2)$的序号是 2，则$[2]_{十进}=[010]_{二进}$，倒置后仍是$[010]_{二进}$，所以 $f(2)\to f(2)$。

又例如，$f(4)$序号为 4，则$[4]_{十进}=[100]_{二进}$，置后是$[001]_{二进}=[1]_{十进}$，所以 $f(1)\to f(4)$。以此类推可求出，$f_1(0)\to f(0)$，$f_1(1)\to f(4)$，$f_1(2)\to f(2)$，$f_1(3)\to f(6)$，$f_1(4)\to f(1)$，$f_1(5)\to f(5)$，$f_1(6)\to f(3)$，$f_1(7)\to f(7)$。这样排序后，第一步运算就构成蝶式运算的方式了。以后，为使后续运算也是蝶式的形式，第二、第三步等也要重新排序。其流程图如图 3-18 所示。

一般说来，用蝶式快速算法需要 $\log_2 N$ 次比特倒置，$2(N-1)$次加减及 N 次乘法。

另外，计算哈尔变换还有一种安德烈亚斯(Andrews)算法，这种算法不是蝶式的。用安德烈亚斯算法需要 $2(N-1)$次加减法及 N 次乘法。

采用蝶式流程算法的目的是使哈尔变换也能用FFT处理机来运算。

二维哈尔变换与二维沃尔什-哈达玛变换完全相似，其中只要把哈达玛矩阵换成哈尔矩阵就可以了。它的定义可用下式表示：

$$[H_a(u,v)] = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cdot \mathrm{har}\left(v,\frac{x}{N}\right)\cdot \mathrm{har}\left(u,\frac{y}{N}\right) \tag{3-181}$$

式中$[f(x,y)]$是空间域数据矩阵，$H_a(u,v)$是变换系数矩阵，$\mathrm{har}\left(v,\frac{x}{N}\right)$和 $\mathrm{har}\left(u,\frac{y}{N}\right)$ 为离散

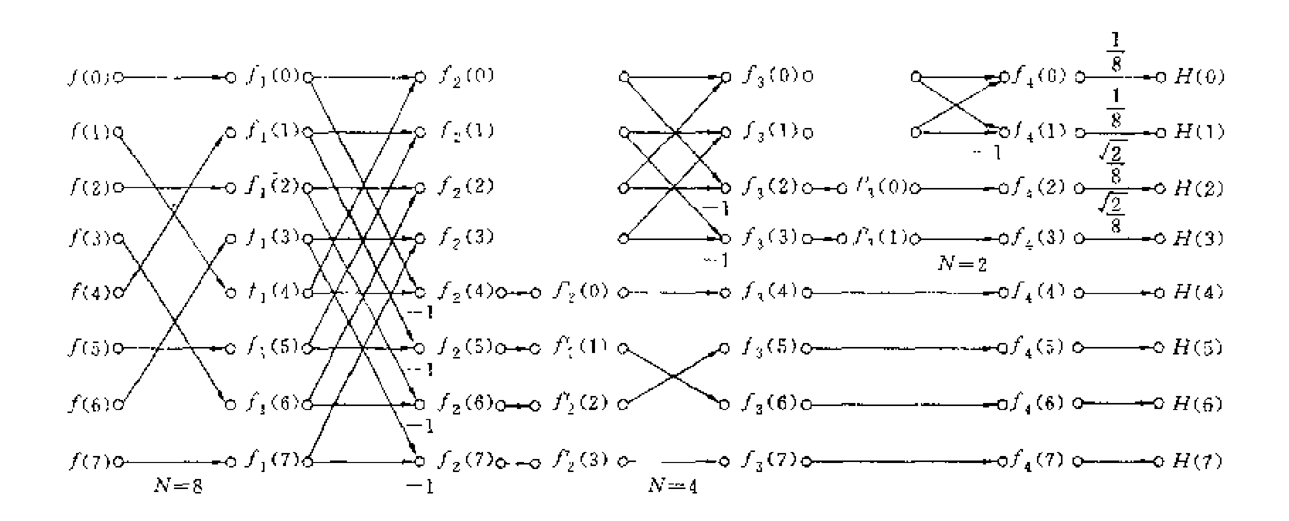


图 3-18 哈尔变换蝶式运算流程图

哈尔函数。其反变换式为

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}H_a(u,v)\cdot \mathrm{har}\left(v,\frac{x}{N}\right)\cdot \mathrm{har}\left(u,\frac{y}{N}\right) \tag{3-182}$$

矩阵式定义如下二式所示：

$$[H_a(u,v)]=\frac{1}{N^2}[\mathrm{har}_{2^p}][f(x,y)][\mathrm{har}_{2^p}]^{-1} \tag{3-183}$$

$$[f(x,y)]=[\mathrm{har}_{2^p}]^{-1}[H_a(u,v)][\mathrm{har}_{2^p}] \tag{3-184}$$

二维哈尔变换的计算也可按一维方法进行。需要指出的一点是，由于哈尔矩阵不是对称矩阵，因此，二维哈尔正变换与逆变换也是不同的。

3.5 斜矩阵与斜变换

在图像处理中要用到的另一种正交变换是斜变换(Slant Transform)。斜向量和斜变换的概念是由伊诺莫托(Enomoto)和夏伊巴塔(Shibata)于 1971 年提出来的。后来关于斜变换的一般定义又由普拉特(Pratt)，韦尔克(Welch)和陈(Chen)进一步推导出来。已经证明，斜向量非常适合于表示灰度逐渐改变的图像信号。目前，斜变换已成功地应用于图像编码。

3.5.1 斜矩阵的构成

斜向量是一个在其范围内呈均匀阶梯下降的离散锯齿波形。$N=4$，阶梯高度为 2 的斜向量如图 3-19 所示。

如果用 $S(n)$ 来表示 $N\times N$ 斜矩阵，设 $N=2^n$，n 为正整数，则

$$S(1)=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \tag{3-185}$$

对于 $N=2^2=4$ 的斜矩阵可以写成下式：

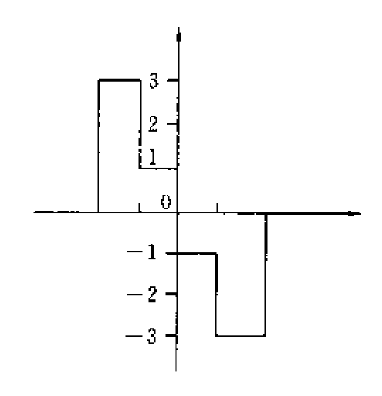


图 3-19　$N=4$，阶梯为 2 的斜向量

$$S(2)=\frac{1}{\sqrt{4}}\begin{bmatrix}1 & 1 & 1 & 1\\ a+b & a-b & -a+b & -a-b\\ 1 & 1 & 1 & 1\\ a-b & -a-b & a+b & -a+b\end{bmatrix} \tag{3-186}$$

式中的 a、b 应满足下列两个条件：第一，阶梯高度必须均匀；第二，$S(2)$必须是正交的。由上述两个条件，我们可以求出 a、b 的值。首先，由阶梯高度必须均匀的条件，可有下式成立：

$$(a+b)-(a-b)=(a-b)-(-a+b)=(-a+b)-(-a-b)$$

即

$$(a+b)-(a-b)=2b$$

$$(a-b)-(-a+b)=2b$$

$$a=2b$$

由此，$S(2)$可写成如下形式：

$$S(2)=\frac{1}{\sqrt{4}}\begin{bmatrix}1 & 1 & 1 & 1\\ 3b & b & -b & -3b\\ 1 & 1 & 1 & 1\\ b & -3b & 3b & -b\end{bmatrix} \tag{3-187}$$

其次，利用正交条件可求出 b 值，

$$\frac{1}{\sqrt{4}}[3b \quad b \quad -b \quad -3b]\cdot\frac{1}{\sqrt{4}}[3b \quad b \quad -b \quad -3b]^{\mathrm{T}}=1$$

$$\frac{1}{4}[9b^2+b^2+b^2+9b^2]=1$$

$$5b^2=1 \qquad\qquad b=\frac{1}{\sqrt{5}}$$

因为 $a=2b$，所以

$$a=\frac{2}{\sqrt{5}}$$

这样便求得斜矩阵如下：

$$S(2)=\frac{1}{\sqrt{4}}\begin{bmatrix}1 & 1 & 1 & 1\\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}}\\ 1 & -1 & -1 & 1\\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}}\end{bmatrix} \tag{3-188}$$

显然，$S(2)$也具有列率性质，$S(2)$的列率为 0，1，1，2。$S(2)$也可以用$S(1)$来表示：

因为

$$S(1)=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

所以，

$$S(2)=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\ a_4 & b_4 & -a_4 & b_4\\ 0 & 1 & 0 & -1\\ -b_4 & a_4 & b_4 & a_4\end{bmatrix}\begin{bmatrix}S(1) & O_2\\ O_2 & S(1)\end{bmatrix} \tag{3-189}$$

其中 $a_4=\frac{2}{\sqrt{5}}$，$b_4=\frac{1}{\sqrt{5}}$。

类似地，$S(3)$用$S(2)$来表示的式子如下：

$$S(3)=\frac{1}{\sqrt{8}}\begin{bmatrix}1 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ a_8 & b_8 & 0 & 0 & -a_8 & b_8 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0\\ -b_8 & a_8 & 0 & 0 & b_8 & a_8 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1\end{bmatrix}\begin{bmatrix}S(2) & O_4\\ O_4 & S(2)\end{bmatrix}$$

$$=\frac{1}{\sqrt{8}}\left[\begin{array}{cccc:cccc}1 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ a_8 & b_8 & 0 & 0 & -a_8 & b_8 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1\\ \hdashline 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0\\ -b_8 & a_8 & 0 & 0 & b_8 & a_8 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1\end{array}\right]$$

$$\times\left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & & & & \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & & O_4 & & \\ 1 & -1 & -1 & 1 & & & & \\ \frac{1}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & & & & \\ \hline & & & & 1 & 1 & 1 & 1 \\ & O_4 & & & \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ & & & & 1 & -1 & -1 & 1 \\ & & & & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{array}\right] \tag{3-190}$$

其中 a_8, b_8 是常数。由上式可见，$S(3)$中的斜向量是通过对 $S(2)$乘以一个比例因子而得到的。其余项是为了满足列率性及正交性而设置的。

由上面二式可把它们推广为$\frac{N}{2}$阶斜矩阵生成 N 阶斜矩阵的公式：

$$S(n)=\frac{1}{\sqrt{2}}\left[\begin{array}{cccc|cccc} 1 & 0 & & & 1 & 0 & & \\ a_N & b_N & & 0 & -a_N & b_N & & 0 \\ & & I_2 & & & & I_2 & \\ & 0 & & I_2 & & 0 & & I_2 \\ \hline 0 & 1 & & & 0 & -1 & & \\ -b_N & a_N & & 0 & b_N & a_N & & 0 \\ & & I_2 & & & & -I_2 & \\ & 0 & & I_2 & & 0 & & -I_2 \end{array}\right] \times\left[\begin{array}{c|c} S(n-1) & 0 \\ \hline 0 & S(n-1) \end{array}\right] \tag{3-191}$$

式中 $\boldsymbol{I}_2=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$为 2×2 单位矩阵；

$$\begin{gathered} a_2 = 1 \\ a_N = 2b_N a_{\frac{N}{2}} \\ b_N = \frac{1}{(1+4a_{\frac{N}{2}}^2)^{\frac{1}{2}}} \\ N = 4,8,16,\cdots,2^n \end{gathered} \tag{3-192}$$

例如，$n=2$ 时，可得

$$b_4 = \frac{1}{(1+4a_2^2)^{\frac{1}{2}}} = \frac{1}{\sqrt{5}}$$

$$a_4 = 2b_4 \cdot a_2 = 2\times\frac{1}{\sqrt{5}} = \frac{2}{\sqrt{5}}$$

代入式(3-191),则

$$S(2)=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}}\\ 0 & 1 & 0 & -1\\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}}\end{bmatrix}\begin{bmatrix}S(1) & 0\\ 0 & S(1)\end{bmatrix}$$

$$=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}}\\ 0 & 1 & 0 & -1\\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}}\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{bmatrix}$$

$$=\frac{1}{\sqrt{4}}\begin{bmatrix}1 & 1 & 1 & 1\\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}}\\ 1 & -1 & -1 & 1\\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}}\end{bmatrix}$$

显然,这样推出的结果与式(3-188)一致。

3.5.2 斜变换

用斜矩阵可定义变换的矩阵式如下:

$$[D(n)]=[S(n)][f(x)] \tag{3-193}$$

式中$[S(n)]$是$N\times N$斜矩阵,并且$N=2^n$,$[D(n)]$是变换系数矩阵,$[f(x)]$是数据矩阵。反变换可由下式表示:

$$[f(x)]=[S(n)]^{-1}[D(n)] \tag{3-194}$$

式中$[S(n)]^{-1}$是$[S(n)]$的逆矩阵。

斜变换也可以用快速算法来计算,下面以$N=4$的情况来说明斜变换的快速算法。

$$S(2)=\frac{1}{\sqrt{4}}\begin{bmatrix}1 & 1 & 1 & 1\\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}}\\ 1 & -1 & -1 & 1\\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}}\end{bmatrix}$$

可以把$S(2)$做如下分解:

$$S(2)=\frac{1}{\sqrt{4}}\begin{bmatrix}1&0&0&0\\0&\frac{3}{\sqrt{5}}&0&0\\0&0&1&0\\0&0&0&\frac{3}{\sqrt{5}}\end{bmatrix}\begin{bmatrix}1&1&0&0\\0&0&1&\frac{1}{3}\\1&-1&0&0\\0&0&\frac{1}{3}&-1\end{bmatrix}\begin{bmatrix}1&0&0&1\\0&1&1&0\\1&0&0&-1\\0&1&-1&0\end{bmatrix}$$

由上面的分解，可得到下面的运算流程图（图 3-20）。

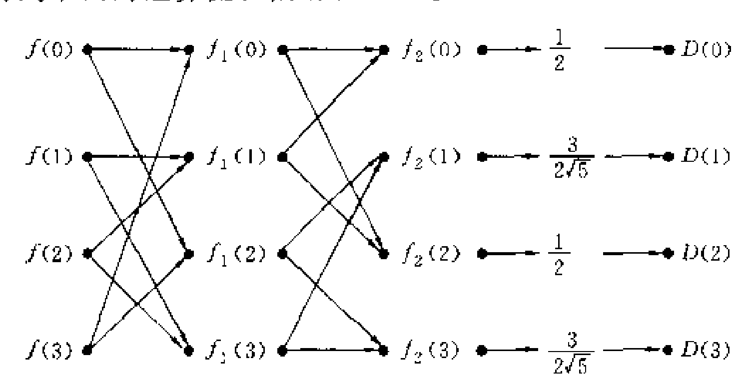


图 3-20　斜变换快速算法流程图

二维斜变换的矩阵式如下：

$$[D(u,v)]=[S(n)][f(x,v)][S(n)]^{-1} \tag{3-195}$$

$$[f(x,y)]=[S(n)]^{-1}[D(u,v)][S(n)] \tag{3-196}$$

式中 $[D(u,v)]$ 是变换系数矩阵，$[f(x,y)]$ 是空间数据矩阵，$[S(n)]$ 是斜矩阵，$[S(n)]^{-1}$ 是 $[S(n)]$ 的逆矩阵。其计算方法也是采用一维计算方法。

以上介绍的是图像处理中应用的几种主要变换法，在某种意义上可以说，这些是比较经典的方法。目前，就变换来说，针对特殊的用途及目的还有更多的变换方法用于图像处理，如数论变换、霍夫变换等。

3.6　小波变换

3.6.1　概述

小波分析是当前应用数学和工程学科中一个迅速发展的新领域，经过近 10 年的探索研究，重要的数学形式化体系已经建立，理论基础更加扎实。与 Fourier 变换、Gabor 变换相比，小波变换是空间（时间）和频率的局部变换，因而能有效地从信号中提取信息。通过伸缩和平移等运算功能可对函数或信号进行多尺度的细化分析，解决了 Fourier 变换不能解决的许多困难问题。小波变换联系了应用数学、物理学、计算机科学、信号与信息处理、图像处理、地震勘探等多个学科。数学家认为，小波分析是一个新的数学分支，它是泛函分析、Fourier 分析、样调分析、数值分析的完美结晶；信号和信息处理专家认为，小波分析是时间-尺度分析和多分辨分析的一种新技术，它在信号分析、语音合成、图像识别、计算机视觉、数据压缩、地震勘探、大气与海洋波分析等方面的研究都取得了有科学意义和应用价值的成果；有人认为，除了微分方程的求解之外，原则上能用 Fourier 分析的地方均可以用小波分析，有时甚至能得到更好的结果。

与 Fourier 变换、Gabor 变换相比，小波变换是时间（空间）频率的局部化分析，它通过伸缩

平移运算对信号(函数)逐步进行多尺度细化,最终达到高频处时间细分,低频处频率细分,能自动适应时频信号分析的要求,从而可聚焦到信号的任意细节,解决了Fourier变换的困难问题,成为继Fourier变换以来在科学方法上的重大突破。有人把小波变换称为"数学显微镜"。

小波分析继承和发展了Gabor变换的局部化思想,基本思想来源于可变窗口的伸缩和平移。其方法的提出可追溯至1910年Haar提出的Haar基,其实这就是最简单的小波基。由于Haar基的不连续性,它没能得到广泛的应用。1936年Littlewood-Pley对Fourier级数建立了L-P理论。L-P理论在频域内有以任意尺度分析函数的能力,可以说L-P理论是多尺度分析的雏形。1952-1962年,Calderon-Zygmund建立了奇异积分算子理论与L-P理论的高维推广,1974年,Coifman对一维H^p空间给出了原子分解,1975年,Calderon用Calderon再生公式给出了抛物型空间H^1的原子分解,它的离散形式已接近于小波展开。Peetre于1976年在使用L-P方法给出Besov空间统一刻画的同时,引出了Besov空间的一组基,其展开系数的大小刻划了Besov空间本身。1981年Stromberg通过对Haar系数的修正,引入了Soblev空间H^s的正交基。实际上这是一组规范化的正交小波基。这些都为小波分析打下了坚实的数学基础。

小波的概念是由法国的从事石油勘测信号处理的地球物理学家Morlet于1984年提出的。他在分析地震波的时频局部特性时,希望使用在高频处时窗变窄,低频处频窗变窄的自适应变换。但Fourier变换很难满足这一要求,随后他引用了高斯余弦调制函数,将其伸缩和平移得到一组函数系,它后来被称之为"Morlet小波基"。Mortet这一根据经验建立的反演公式当时并未得到数学家的认可,幸运的是Calderon的发现、Hardy空间原子分解的深入研究已为小波变换的诞生作了理论上的准备。后来,Stromberg构造了第一个小波基。1986年著名的数学家Meyer构造了一个真正的小波基,并与Mallat合作建立了构造小波基的统一方法——多尺度分析。从此,小波分析开始了蓬勃发展的阶段。值得一提的是,比利时女数学家Daubechies的《Ten Lectures on Wavelet》(小波十讲)一书对小波的普及应用起了重要的推动作用。

1986年Jafferd、Lemavie和Meyer与从事信号处理的Mallat合作指出小波正交基的构造可纳入一个统一框架,引入多分辨分析的概念,统一了前人构造的具体小波,并给出了多分辨分析的构造正交小波基的一般化方法。Mallat还在Brut和Adelson的塔式分解算法启发下,提出了小波变换的快速分解与重构算法,现在称之为Mallat算法。该算法在小波分析中的作用相当于FFT在Fourier变换中的地位。后来,该方法成功地应用于图像的分解与重建。在用于时频分析时,希望小波具有紧支撑集,1988年Daubechies首先构造了紧支集光滑小波。从此,小波分析理论得到了系统化。

虽然小波正交基用途广泛,但也存在着不足。其一是小波正交基的结构复杂,其次具有紧支集的小波正交基不可能具有对称性。因此,它用作滤波器不可能有线性相位,从而产生信号重构失真。为解决此类问题又产生了所谓双正交小波基理论。在实际应用中,人们还构造了周期小波和多元小波等。近年来小波分析已深入到非线性逼近、统计信号处理等领域。由此可见,小波理论及应用正在逐步发展与完善。随着理论及算法的成熟,小波分析必将有较大的作为。

3.6.2 时-频分析

信号分析的主要目的是寻找一种简单有效的信号变换方法,以便突出信号中的重要特性,简化运算的复杂度。大家熟知的Fourier变换就是一种刻画函数空间,求解微分方程,进行数值计算的主要方法和有效的数学工具。它可把许多常见的微分、积分和卷积运算简化为代数运算。从物理意义上理解,一个周期振动信号可看成是具有简单频率的简谐振动的叠加。Fourier

展开正是这一物理过程的数学描述。即：

$$F(\omega)=\int_{-\infty}^{+\infty}f(t)e^{-j\omega t}dt \tag{3-197}$$

$$f(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty}F(\omega)e^{j\omega t}d\omega \tag{3-198}$$

Fourier 变换的特点是域变换，它把时域和频域联系起来，把时域内难以显现的特征在频域中十分清楚地显现出来。频谱分析的本质就是对 $F(\omega)$ 的加工和处理。基于这一基本原理，现代谱分析已研究与发展了多种行之有效的高效、多分辨率的分析算法。

在实际过程中，时变信号是常见的，如语音信号、地震信号、雷达回波等。在这些信号的分析中，希望知道信号在突变时刻的频率成份，显然利用 Fourier 变换处理这些信号，这些非平稳的突变成份往往被 Fourier 变换的积分作用平滑掉了。由于 $|e^{+j\omega t}|=1$，因此，频谱 $F(\omega)$ 的任一频率成份的值是由时域过程 $f(t)$ 在 $(-\infty,+\infty)$ 上的贡献决定的，而过程 $f(t)$ 在任一时刻的状态也是由 $F(\omega)$ 在整个频域 $(-\infty,+\infty)$ 的贡献决定的。该性质可由熟知的 $\delta(t)$ 函数来理解，即时域上的一个冲激脉冲在频域中具有无限伸展的均匀频谱。$f(t)$ 与 $F(\omega)$ 间的彼此的整体刻画，不能反映各自在局部区域上的特征。因此，不能用于局部分析。在实际应用中，也不乏不同的时间过程却对应着相同的频谱的例子。

1. Gabor 变换

由于 Fourier 变换存在着不能同时进行时间-频率局部分析的缺点，曾出现许多改进的方法。1946 年 Gabor 提出一种加窗的 Fourier 变换方法，它在非平稳信号分析中起到了很好的作用。是一种有效的信号分析方法，而且与当今的小波变换有许多相似之处。

(1) Gabor 变换的定义

在 Fourier 变换中，把非平稳过程看成是一系列短时平稳信号的叠加，而短时性是通过时间上加窗来实现的。整个时域的覆盖是由参数 τ 的平移达到的。换句话说，该变换是用一个窗函数 $g(t-\tau)$ 与信号 $f(t)$ 相乘实现在 τ 附近开窗和平移，然后施以 Fourier 变换，这就是 Gabor 变换也称短时 Fourier 变换或加窗 Fourier 变换。Gabor 变换的定义由下式给出：对于 $f(t)\in L^2(R)$，

$$Gf(\omega,\tau)=\int_{-\infty}^{+\infty}f(t)g(t-\tau)e^{-j\omega t}dt \tag{3-199}$$

其中 $g(t-\tau)e^{-j\omega t}$ 是积分核。该变换在 τ 点附近局部测量了频率为 ω 的正弦分量的幅度。通常 $g(t)$ 选择能量集中在低频处的实偶函数；Gabor 采用高斯(Gauss)函数作窗的函数，相应的 Fourier 变换仍旧是 Gauss 函数，从而保证窗口 Fourier 变换在时域和频域内均有局部化功能。

令窗口函数为 $g_a(t)$ 则有

$$g_a(t)=\frac{1}{2\sqrt{\pi a}}e^{-t^2/4a} \tag{3-200}$$

式中 a 决定了窗口的宽度，$g_a(t)$ 的 Fourier 变换用 $G_a(\omega)$ 表示，则有

$$\begin{aligned}G_a(\omega)&=\int_{-\infty}^{+\infty}g_a(t)e^{-j\omega t}dt\\&=\int_{-\infty}^{+\infty}\frac{1}{2\sqrt{\pi a}}e^{-\frac{t^2}{4a}}e^{-j\omega t}dt\\&=\frac{1}{2\sqrt{\pi a}}\int_{-\infty}^{+\infty}e^{-\frac{t^2}{4a}}e^{-j\omega t}dt\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2\sqrt{\pi a}} \int_{-\infty}^{+\infty} \mathrm{e}^{-\left(\frac{t^2}{4a} + \mathrm{j}\omega t\right)} \mathrm{d}t \\
&= \frac{1}{2\sqrt{\pi a}} \cdot \sqrt{4\pi a} \mathrm{e}^{-a\omega^2} = \mathrm{e}^{-a\omega^2}
\end{aligned} \tag{3-201}
$$

由此可得到：

$$
\begin{aligned}
\int_{-\infty}^{+\infty} Gf(\omega,\tau)\mathrm{d}\tau &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(t) g_a(t-\tau) \cdot \mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t\mathrm{d}\tau \\
&= \int_{-\infty}^{+\infty} f(t)\mathrm{e}^{-\mathrm{j}\omega t}\int_{-\infty}^{+\infty} g_a(t-\tau)\mathrm{d}t\mathrm{d}\tau \\
&= \int_{-\infty}^{+\infty} f(t)\mathrm{e}^{-\mathrm{j}\omega t}\left(\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{\pi a}} \mathrm{e}^{\frac{-(t-\tau)^2}{4a}}\mathrm{d}\tau\right)\mathrm{d}t \\
&= \int_{-\infty}^{+\infty} f(t)\mathrm{e}^{-\mathrm{j}\omega t}\left(\int_{-\infty}^{+\infty} \frac{1}{2\sqrt{\pi a}} \mathrm{e}^{\frac{-u^2}{4a}}\mathrm{d}u\right)\mathrm{d}t \\
&= \int_{-\infty}^{+\infty} f(t)\mathrm{e}^{-\mathrm{j}\omega t}\left(\frac{1}{2\sqrt{\pi a}} \sqrt{4\pi a}\right)\mathrm{d}t \\
&= \int_{-\infty}^{+\infty} f(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t = F(\omega)
\end{aligned} \tag{3-202}
$$

显然，信号 $f(t)$ 的 Gabor 变换按窗口宽度分解了 $f(t)$ 的频谱 $F(\omega)$，提取出它的局部信息。当 τ 在整个时间轴上平移时，就给出了 Fourier 的完整变换。

相应的重构公式为

$$
f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} G_\tau(\omega) g(t-\tau)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}\omega\mathrm{d}\tau \tag{3-203}
$$

窗口 Fourier 变换是能量守恒变换，即

$$
\int_{-\infty}^{+\infty} |f(t)|^2\mathrm{d}t = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} |G_\tau(\omega)|^2\mathrm{d}\omega\mathrm{d}\tau \tag{3-204}
$$

这里应注意，积分核 $g(t)\mathrm{e}^{-\mathrm{j}\omega t}$ 对所有 ω 和 τ 都有相同的支撑区，但周期数随 ω 而变化。支撑区是指一个函数或信号 $f(t)$ 的自变量 t 的定义域，当 t 在定义域内取值时 $f(t)$ 的值域不为零，在支撑区之外信号或过程下降为零。

为了研究窗口 Fourier 变换的时频局部化特性就要研究 $|g_{\omega,\tau}|^2$ 和 $|G_{\omega,\tau}|^2$ 的特性。这里 $G_{\omega,\tau}$ 是 $g_{\omega,\tau}$ 的 Fourier 变换。由于 Fourier 变换是能量守恒的，所以，有 Parseval 定理存在。即：

$$
\int_{-\infty}^{+\infty} f(t)\overline{g}_{\omega,\tau}(t)\mathrm{d}t = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)\overline{G}_{\omega,\tau}(\omega)\mathrm{d}\omega \tag{3-205}
$$

这里的 $\overline{g}_{\omega,\tau}(t)$ 和 $\overline{G}_{\omega,\tau}(\omega)$ 分别是 $g_{\omega,\tau}(t)$ 和 $G_{\omega,\tau}(\omega)$ 的复共轭函数，当为实数时两种表示是相等的。如果把上述函数乘积的积分运算用内积符号表示，则有

$$
\langle f, y\rangle = \int_{-\infty}^{+\infty} f(x)\overline{y}(x)\mathrm{d}x \qquad f, y \in L^2(R) \tag{3-206}
$$

其中 f 和 y 都是在实数域的平方可积函数。

由此，

$$
\langle f, g_{\omega,\tau}\rangle = \frac{1}{2\pi}\langle G_{\omega,\tau}, F(\omega)\rangle \tag{3-207}
$$

当 $f(x) = y(x)$ 则有

$$
\langle f, f\rangle = \int_{-\infty}^{+\infty} |f(x)|^2\mathrm{d}t = \| f(x) \|^2 \tag{3-208}
$$

符号 $\|f(x)\|$ 叫做 $f(x)$ 的范数。

这一表达式的物理意义是 Fourier 变换的时域(t)和频域(ω)的一对共轭变量(ω,t)具有对易关系,从而使 Fourier 变换与加窗口的 Fourier 变换具有对称性。

如果用角频率变量 γ 代替时间变量 t,用频域窗口函数 $G(\gamma,\omega)$ 代替时域窗口函数 $g(t-\tau)$ 则可得到:

$$
\begin{aligned}
Gf(\omega,\tau) &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\gamma)\,\overline{[G(\gamma-\omega)\mathrm{e}^{-\mathrm{j}\omega\tau}]\mathrm{e}^{-\mathrm{j}\gamma\tau}}\mathrm{d}\gamma \\
&= \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\gamma)\overline{G}_{\omega,\tau}(\gamma)\mathrm{e}^{-\mathrm{j}\gamma\tau}\mathrm{d}\gamma
\end{aligned} \tag{3-209}
$$

这里 $G_{\omega,\tau}(\gamma)$ 是时域窗口函数 $g_{\omega,\tau}(t)$ 的 Fourier 变换。该式的意义在于频域中的信号 $F(\gamma)$ 通过窗口函数 $G_{\omega,\tau}(\gamma)$ 的加窗作用获得了 $F(\gamma)$ 在频率 ω 附近的局部信息,即

$$
F(\omega) = \overline{G}(\gamma-\omega)\cdot F(\gamma) \tag{3-210}
$$

如果选用窗口函数在时域和频域均有良好的局部性质,那么可以说 Fourier 变换给出了信号 $f(t)$ 的局部时-频分析。这样就有利于同时在频域和时域提取信号 $f(t)$ 的精确信息。

(2) Heisenberg 测不准原理

由 Gabor 变换的局部时-频分析的原理,人们自然希望得到合适的时窗和频窗选择方法,以便提取精确的信息,这自然涉及到窗口的选择问题。

如果我们把 $|g(t)|^2$ 和 $|G(\omega)|^2$ 作为窗口函数在时域和频域的重量分布,令 t_0 和 ω_0 分别表示时窗和频窗的"重心", σ_{g_t} 和 σ_{G_ω} 表示 $g_{\omega,\tau}(t)$ 和 $G_{\omega,\tau}(\omega)$ 的均方差,则有

$$
\begin{aligned}
t_0 &= \frac{1}{\|g(t)\|^2}\int_{-\infty}^{+\infty} t|g(t)|^2\mathrm{d}t \\
\omega_0 &= \frac{1}{\|G(\omega)\|^2}\int_{-\infty}^{+\infty}\omega|G(\omega)|^2\mathrm{d}\omega
\end{aligned} \tag{3-211}
$$

如果作归一化处理,令 $\|g(t)\|^2 = \frac{1}{2\pi}\|G(\omega)\|^2 = 1$

则上式为

$$
t_0 = \int_{-\infty}^{+\infty} t|g(t)|^2\mathrm{d}t \tag{3-212}
$$

$$
\omega_0 = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\omega|G(\omega)|^2\mathrm{d}\omega
$$

$$
\sigma_{g_t}^2 = \frac{1}{\|g(t)\|^2}\int_{-\infty}^{+\infty}(t-t_0)|g(t)|^2\mathrm{d}t = \int_{-\infty}^{+\infty}(t-t_0)^2|g(t)|^2\mathrm{d}t
$$

$$
\sigma_{G_\omega} = \frac{1}{\|G(\omega)\|^2}\int_{-\infty}^{+\infty}(\omega-\omega_0)|G(\omega)|^2\mathrm{d}\omega = \frac{1}{2\pi}\int_{-\infty}^{+\infty}(\omega-\omega_0)^2|G(\omega)|^2\mathrm{d}\omega \tag{3-213}
$$

这里我们把 σ_{g_t} 和 σ_{G_ω} 作为衡量时-频局部特性的一个标准。设 $t_0=0$, $\omega_0=0$,则有

$$
\sigma_{g_t}^2\cdot\sigma_{G_\omega}^2 = \int_{-\infty}^{+\infty} t^2|g(t)|^2\mathrm{d}t\cdot\frac{1}{2\pi}\int_{-\infty}^{+\infty}\omega^2|G(\omega)|^2\mathrm{d}\omega \tag{3-214}
$$

对于窗口函数,假设 $\lim\limits_{|t|\to\infty} t|g(t)|^2 = 0$

并且 $g(t)$ 的导数 $g'(t)$ 的 Fourier 变换为 $\omega(G\omega)$,所以,

$$
\frac{\mathrm{d}}{\mathrm{d}t}(t|g(t)|^2) = |g(t)|^2 + \frac{\mathrm{d}}{\mathrm{d}t}|g(t)|^2 = |g(t)|^2 + 2tg'(t)g(t) = 0
$$

即：

$$tg(t)g'(t) = -\frac{1}{2}|g(t)|^2$$

由此，

$$\begin{aligned}\sigma_{g_t}^2 \cdot \sigma_{G_\omega}^2 &= \int_{-\infty}^{+\infty} t^2 |g(t)|^2 \mathrm{d}t \cdot \frac{1}{2\pi}\int_{-\infty}^{+\infty} \omega^2 |G(\omega)|^2 \mathrm{d}\omega \\ &= \left(\int_{-\infty}^{+\infty} |tg(t)|^2 \mathrm{d}t\right) \cdot \frac{1}{2\pi}\int_{-\infty}^{+\infty} 2\pi |g'(t)|^2 \mathrm{d}t \geqslant \left|\int_{-\infty}^{+\infty} tg(t)g'(t)\mathrm{d}t\right|^2 \\ &= \left|\int_{-\infty}^{+\infty} -\frac{1}{2}|g(t)|^2 \mathrm{d}t\right|^2 = \frac{1}{4}\left(\int_{-\infty}^{+\infty} |g(t)|^2 \mathrm{d}t\right)^2 = \frac{1}{4}\end{aligned} \tag{3-215}$$

所以，

$$\sigma_{g_t} \cdot \sigma_{G_\omega} \geqslant \frac{1}{2} \tag{3-216}$$

这说明在对信号作时-频分析时，其时窗和频窗不能同时达到极小值，这就是 Heisenberg 测不准原理。上式如果用 f 代替角频率 $f=\frac{\omega}{2\pi}$，则

$$\sigma_{g_t} \cdot \sigma_{G_\omega} \geqslant \frac{1}{4\pi} \tag{3-217}$$

由上式可知，无论是周期信号还是非周期信号，既可以用 $f(t)$ 来描述也可以用 $F(\omega)$ 来描述，采用哪种方法可根据具体问题而定。但如果对时域和频域均感兴趣的话，Gabor 变换是一个有力的分析工具。但它受到 Heisenberg 测不准原理的制约。要想同时达到时间 t 与频率 ω 的最高分辨率是不可能的。

对于时-频窗口的选择应遵循如下规律：对于宽带信号，由于频率变化剧烈，为了能准确提取高频信息需要有足够的时间分辨率，t_0 应选小值，可这样会造成样本点多、计算量大，难于获得快速高效算法。为了提取高频分量，时域窗口应尽量窄，频域窗口适当放宽。对于慢变的低频信号，时窗可适当加宽，而频窗应尽量缩小，保证有较高的频率分辨率和较小的测量误差。总之，对多尺度信号希望时-频窗口有自适应性，自动改变 σ_{g_t} 和 σ_{G_ω} 的大小。高频情况下，频窗大、时窗小；低频情况下，频窗小、时窗大。

但 Gabor 变换的时-频窗口是固定不变的，窗口没有自适应性，不适于分析多尺度信号过程和突变过程，而且其离散形式没有正交展开，难于实现高效算法，这是 Gabor 变换的主要缺点，因此也就限制了它的应用。

3.6.3 连续小波变换

在信号分析与处理中，为了提高算法的效率，对信号进行变换处理的积分核应属于正交基，作为信号分析的有效数学工具的标志应是可变窗口、平移功能和正交性。Gabor 提出的 Gabor变换为此迈出了关键的一步，因为 Gabor 变换已具备了平移功能，只是其相当于放大倍数固定的显微镜而已。在这方面 Morlet 为此作出了重大贡献。

1. 小波变换的定义

设函数 $f(t)$ 具有有限能量，即 $f(t)\in L^2(R)$

则小波变换的定义如下：

$$W_f(a,b) = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t)\mathrm{d}t = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)\mathrm{d}t \tag{3-218}$$

$$a > 0,\quad f(t) \in L^2(R)$$

积分核为 $\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$ 的函数族。其中 a 为尺度参数，b 为定位参数，函数 $\psi_{a,b}(t)$ 称为小波。如果 $\psi_{a,b}(t)$ 是复变函数时，上式采用复共轭函数 $\overline{\psi}_{a,b}(t)$，若 $a>1$ 函数 $\psi(t)$ 具有伸展作用，$a<1$ 函数具有收缩作用。而 Fourier 变换 $\Psi(\omega)$ 则恰好相反。伸缩参数 a 对小波 $\psi(t)$ 的影响见图 3-21。小波 $\psi_{a,b}(t)$ 随伸缩参数 a 平移参数 b 而变化如图 3-22 所示。



图 3-21　伸缩参数 a 对生成小波 $\psi(t)$ 的影响

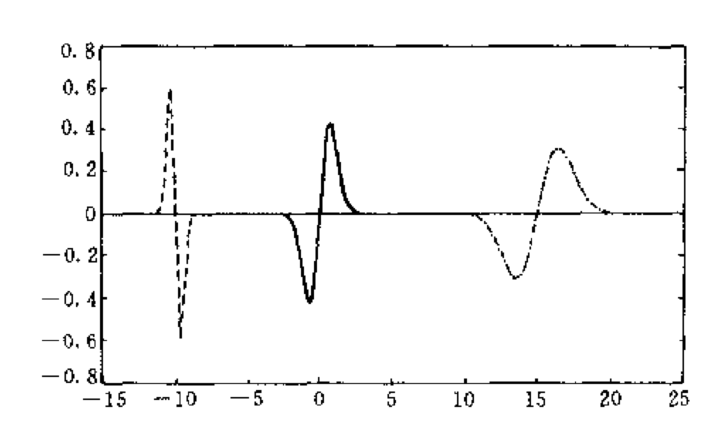


图 3-22　小波 $\psi_{a,b}(t)$ 的波形随参数 a,b 变化的情形

图中小波函数为 $\psi(t)=te^{-t^2}$。当 $a=2,b=15$ 时，$\psi_{a,b}(t)=\psi_{2,15}(t)$ 的波形 $\psi(t)$ 从原点向右移至 $t=15$ 且波形展宽，$a=-5$，$b=-10$ 时，$\psi_{-5,-10}(t)$ 则是 $\psi(t)$ 从原点向左平移至 $t=-10$ 处且波形收缩。

随着参数 a 的减小，$\psi_{a,b}(t)$ 的支撑区也随之变窄，而 $\Psi_{a,b}(\omega)$ 的频谱随之向高频端展宽，反之亦然。这就有可能实现窗口大小自适应变化，当信号频率增高时，时窗宽度变窄，而频窗宽度增大，有利于提高时域分辨率，反之亦然。

小波 $\psi(t)$ 的选择既不是唯一的，也不是任意的。这里 $\psi(t)$ 是归一化的具有单位能量的解析函数，它应满足如下几个条件：

1）定义域应是紧支撑的(Compact support)，换句话说就是在一个很小的区间之外，函数为零，也就是函数应有速降特性。

2）平均值为零，即：

$$\int_{-\infty}^{\infty}\psi(t)\mathrm{d}t = 0, \tag{3-219}$$

其高阶矩也为零。

$$\int_{-\infty}^{\infty} t^k\psi(t)\mathrm{d}t = 0 \qquad k = 0,1,2,\cdots,N-1 \tag{3.220}$$

该条件也叫小波的容许条件(Admissibility condition)。

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty \tag{3-221}$$

式中 $\Psi(\omega) = \int_{-\infty}^{+\infty} \psi(t) e^{-j\omega t} dt$，$C_\psi$ 是有限值，它意味着 $\Psi(\omega)$ 连续可积，

$$\Psi(0) = \int_{-\infty}^{+\infty} \psi(t) dt = 0 \tag{3-222}$$

由上式可以看出，小波 $\psi(t)$ 在 t 轴上取值有正有负才能保证式(3-222)积分为零。所以，$\psi(t)$ 应有振荡性。

上面两个条件可概括为，小波应是一个具有振荡性和迅速衰减的波。

对于所有的 $f(t),\psi(t) \in L^2(R)$，连续小波逆变换由式(3-223)给出：

$$f(t) = \frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} a^{-2} W_f(a,b)\psi_{a,b}(t) \mathrm{d}a\mathrm{d}b \tag{3-223}$$

逆变换公式可证明如下：

$$\Psi_{a,b}(\omega) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} \psi\left(\frac{t-b}{a}\right) e^{-j\omega t} dt = e^{-j\omega b}\sqrt{|a|}\,\Psi(a\omega)$$

$$W_f(a,b) = \langle f,\psi_{a,b}\rangle = \frac{1}{2\pi}\langle F,\psi_{a,b}\rangle$$

$$= \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)\cdot \overline{\Psi}_{a,b}(\omega) d\omega$$

$$= \frac{\sqrt{|a|}}{2\pi}\int_{-\infty}^{+\infty} F(\omega)\cdot e^{-j\omega b}\Psi(a\omega) d\omega$$

由此可得：

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} a^{-2} W_f(a,b)\overline{W}_g(a,b) \mathrm{d}a\mathrm{d}b$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left\{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{|a|}{4\pi^2}F(\omega)\overline{G}(\omega')e^{j\omega b}e^{-j\omega' b}\cdot\Psi(a\omega)\,\overline{\Psi}(a\omega')d\omega d\omega'\right\}\frac{1}{a^2}\mathrm{d}a\mathrm{d}b$$

$$= \frac{1}{4\pi^2}\int_{-\infty}^{+\infty} d\omega\int_{-\infty}^{+\infty}\overline{\Psi}(a\omega')F(\omega)\frac{\mathrm{d}a}{|a|}\int_{-\infty}^{+\infty} e^{-j\omega b}\mathrm{d}b\int_{-\infty}^{+\infty} e^{j\omega' b}\,\overline{\Psi}(a\omega')\overline{G}(\omega')d\omega'$$

$$= \frac{1}{2\pi}\int_{-\infty}^{+\infty} d\omega\int_{-\infty}^{+\infty}\overline{\Psi}(a\omega)F(\omega)\phi(a\omega)\overline{G}(\omega)\frac{\mathrm{d}a}{|a|}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{+\infty} d\omega\int_{-\infty}^{+\infty}|\Psi(a\omega)|^2F(\omega)\overline{G}(\omega)\frac{\mathrm{d}a}{|a|}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)\overline{G}(\omega)d\omega\int_{-\infty}^{+\infty}\frac{|\Psi(a\omega)|^2}{|(a\omega)|}\mathrm{d}a\omega = C_\psi\langle f,g\rangle \tag{3-224}$$

如果 $g(t)$ 为 Gauss 分布，

$$g(t) = \frac{1}{2\sqrt{\pi\sigma}}e^{\frac{-(t-t_0)^2}{4\sigma}} \tag{3-225}$$

当 $\sigma\to0^+$ 时，$g(t)\to\delta(t-t_0)$。由于 $\delta(t)$ 的取样特性，即

$$\int_{-\infty}^{+\infty} f(t)\delta(t-t_0)\mathrm{d}t = f(t_0)$$

这里 t_0 是任意值并可连续变化，因此：

$$\lim_{\sigma\to0}\langle f, g_\sigma\rangle = \langle f,\lim_{\sigma\to0} g_\sigma\rangle = \langle f,\delta(\cdot - t)\rangle = f(t)$$

所以，

$$\begin{aligned}
&\lim_{\sigma\to 0^+} C_\psi\langle f,\ g_\sigma\rangle = C_\psi \lim_{\sigma\to 0^+}\langle f,\ g_\sigma\rangle = C_\psi\langle f,\ \sigma(\cdot - t)\rangle = C_\psi f(t)\\
&\lim_{\sigma\to 0^+}\overline{W_g(a,b)} = \lim_{\sigma\to 0^+}\overline{\langle g,\ \psi_{a,b}\rangle} = \langle\delta(\cdot - t),\psi_{a,b}\rangle = \psi_{a,b}(t)
\end{aligned} \tag{3-226}$$

这里 $\sigma(\cdot - t)$中的“$\cdot$”表示任意变量。

代入式(3-223)得到：

$$f(t) = \frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} W_f(a,b)\psi_{a,b}(t)\,\frac{\mathrm{d}a\mathrm{d}b}{a^2} \tag{3-227}$$

因此，逆变换得证。该式也说明 $f(t)$的小波变换是能量守恒的。即有下式成立：

$$\int_{-\infty}^{+\infty}|f(t)|^2\mathrm{d}t = \frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|W_f(a,b)|^2\,\frac{\mathrm{d}a\mathrm{d}b}{a^2} \tag{3-228}$$

2. 小波变换的时-频局部化

小波 $\psi_{a,b}(t)$是紧支撑函数，小波变换实现时-频局部化分析的特点与信号频率高低密切相关。因此，了解小波变换在频域中的特性是很重要的。由能量守恒定理可知：

$$W_f(a,b) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}F(\omega)\cdot\Psi_{a,b}(\omega)\mathrm{d}\omega \tag{3-229}$$

为了解小波变换的时-频局部化特点，我们同样需要了解$|\psi_{a,b}(t)|^2$与$|\Psi_{a,b}(\omega)|^2$的均方差 $\sigma_{\psi_{a,b}}$，$\sigma_{\Psi_{a,b}}$。这里我们定义：

$$\begin{aligned}
\sigma_{\Psi_{a,b}} &= \left\{\int_{-\infty}^{+\infty}(\omega - \omega^0_{\Psi_{a,b}})^2|\ \Psi_{a,b}(\omega)\,|^2\mathrm{d}\omega\right\}^{\frac{1}{2}}\\
\sigma_{\psi_{a,b}} &= \left\{\int_{-\infty}^{+\infty}(t - t_0)\,|\ \psi_{a,b}(t)\,|^2\mathrm{d}t\right\}^{\frac{1}{2}}
\end{aligned} \tag{3-230}$$

其中

$$\omega^0_{\Psi_{a,b}} = \frac{\int_0^{+\infty}\omega|\Psi_{a,b}(\omega)|^2\mathrm{d}\omega}{\int_0^{+\infty}|\Psi_{a,b}(\omega)|^2\mathrm{d}\omega}$$

$$t_0 = \frac{\int_{-\infty}^{+\infty}t\,|\psi_{a,b}(t)|^2\mathrm{d}t}{\int_{-\infty}^{+\infty}|\psi_{a,b}(t)|^2\mathrm{d}t}$$

如果 $a=1$，$b=0$ 时容易推出：

$$\begin{aligned}
\sigma_{\Psi_{a,b}} &= \frac{\sigma_{\Psi_{1,0}}}{a}\\
\omega^0_{\Psi_{a,b}} &= \frac{\omega_{\Psi_{1,0}}}{a}
\end{aligned} \tag{3-231}$$

由上述关系显见，$\omega_{\Psi_{a,b}}$随函数的伸展而变小，即带通的中心向低频分量偏移；反之，$\omega_{\Psi_{a,b}}$随 a 的减小而变大，带通中心向高频分量偏移。在小波变换中时间窗口的宽度与频率窗口的宽度是尺度参数 a 的函数，但其乘积由 Heisenberg 测不准原理限定为一常数，因此，高频分量在时域局部化分辨率提高是以频域的不确定性加大换取的。分析高频分量时，时窗变窄，频窗加宽，分析低频分量时时窗变宽，频窗变窄，从而实现了时-频窗口的自动自适应变化。图 3-23 示出了加窗的 Fourier 分析和小波分析的时频特性比较，由图可见，(a)加窗的 Fourier 分析的基

函数振荡个数不同，而小波分析的基函数常具有数个振荡；加窗的 Fourier 分析的时频分辨率固定，而小波分析的时频分辨率可变。

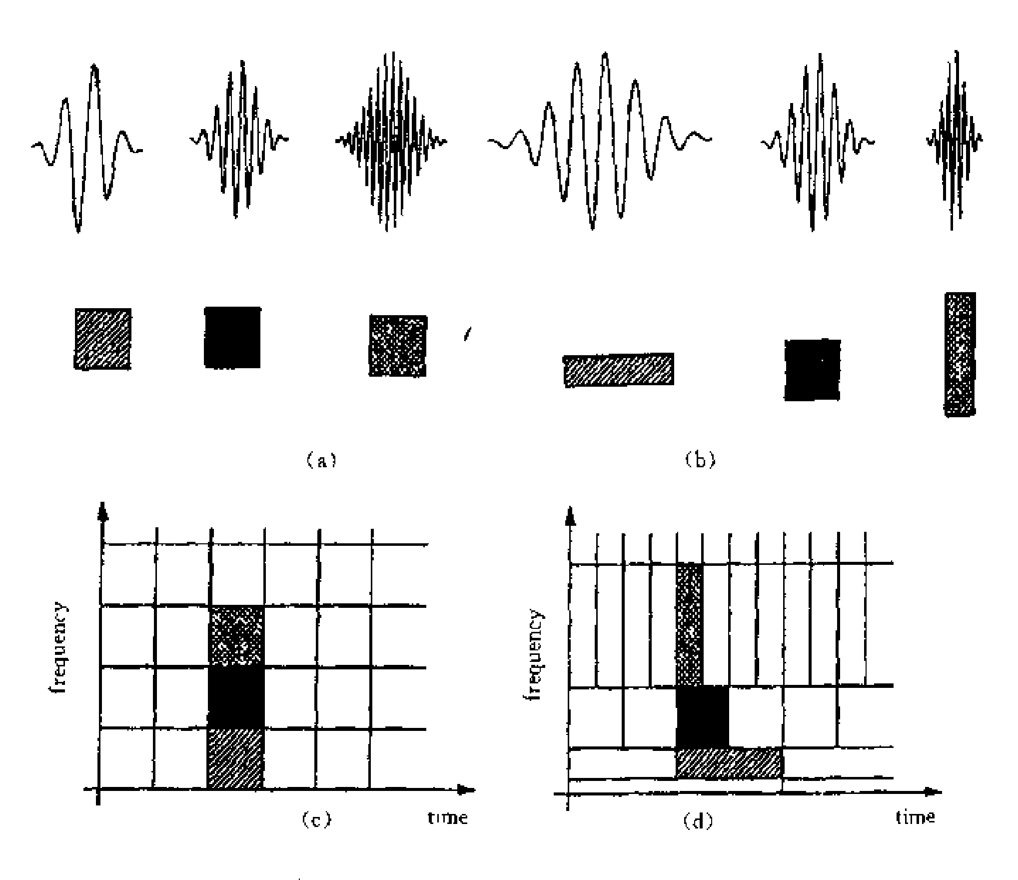


图 3-23 加窗 Fourier 分析和小波分析的时频特性比较

图 3-24 显示了 Gabor 变换与小波变换的滤波特性。由图可见 Gabor 滤波是恒定带宽滤波，而小波滤波随着中心频率增加而带宽加大。

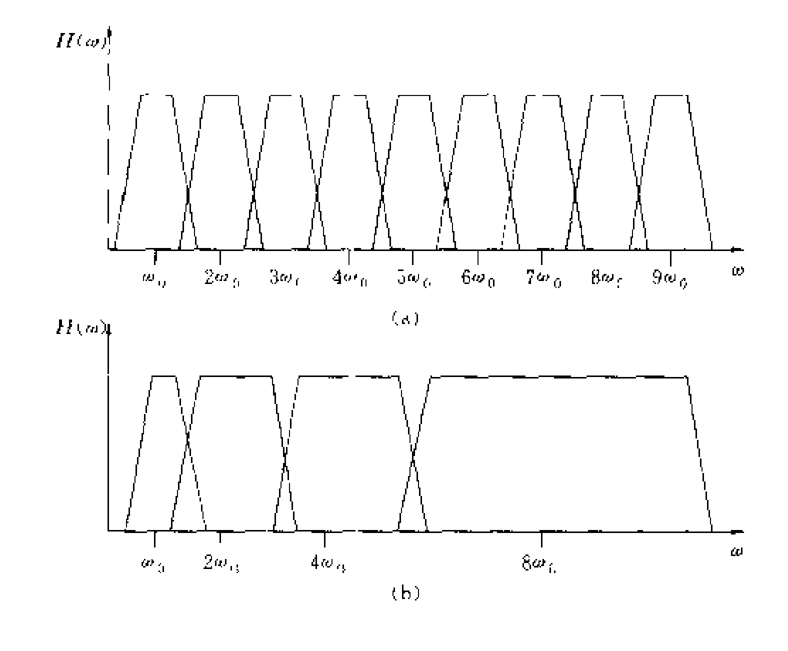


图 3-24 窗口 Fourier(Gabor)变换特性(a)和小波滤波特性(b)

3. 几种典型的一维小波

小波的选择是灵活的，凡能满足条件的函数均可作为小波函数，这里仅介绍几种具有代表性的小波以供参考。

(1) Haar 小波

$$\psi_H(t)=\begin{cases}1 & 0\leqslant t<\frac{1}{2}\\ -1 & \frac{1}{2}\leqslant t<1\\ 0 & 其他\end{cases}\tag{3-232}$$

该正交函数是由 Haar 于 1910 年提出的，对 t 平移则可得到：

$$\int_{-\infty}^{+\infty}\psi_H(t)\psi_H(t-n)\mathrm{d}t=0\qquad n=0,\pm 1,\pm 2,\cdots\tag{3-233}$$

其波形如图 3-25 所示。

(2) Mexico Hat 小波

Mexico Hat 小波是 Gauss 函数的二阶导数，即：

$$\psi(t)=\frac{2}{\sqrt{3}}\pi^{-\frac{1}{4}(1-t^2)}\mathrm{e}^{-\frac{t^2}{2}}\tag{3-234}$$

Mexico Hat 小波也叫 Marr 小波，Mexico Hat 小波是实值小波，它的更普遍的形式由下式给出，即由 Gauss 分布的 n 阶导数给出：

$$\psi_n(t)=(-1)^n\frac{\mathrm{d}^n}{\mathrm{d}t^n}(\mathrm{e}^{-\frac{|t|^2}{2}})\tag{3-235}$$

相应的谱为

$$\Psi_n(\omega)=n(\mathrm{j}\omega)^n\mathrm{e}^{-\frac{|\omega|^2}{2}}\tag{3-236}$$

其波形如图 3-26 所示。

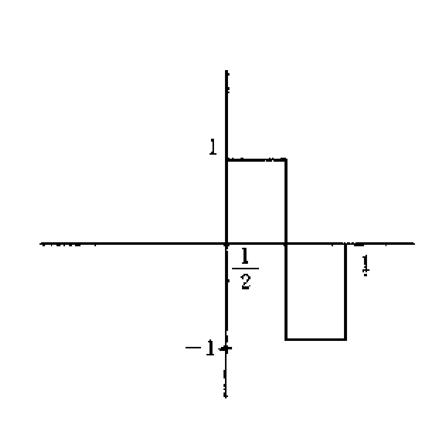


图 3-25 Haar 小波

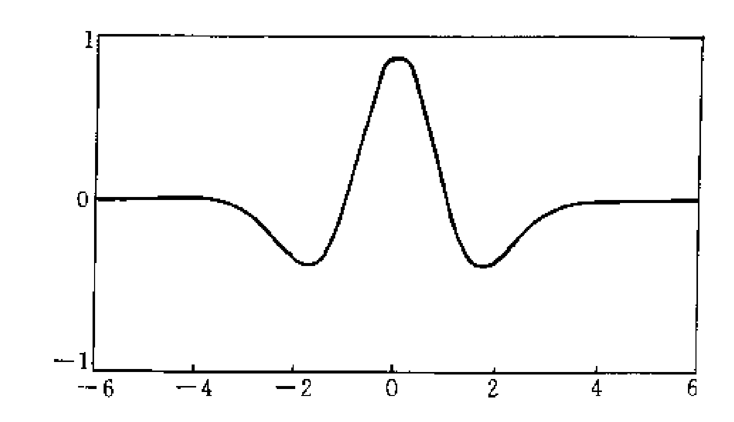


图 3-26 Mexico Hat 小波

(3) Morlet 小波

Morlet 小波是最常用的复值小波，它可由下式给出：

$$\psi(t)=\pi^{-\frac{1}{4}}\left[\mathrm{e}^{-\mathrm{j}\omega_0 t}-\mathrm{e}^{-\frac{\omega_0^2}{2}}\right]\mathrm{e}^{-\frac{t^2}{2}}\tag{3-237}$$

其 Fouvier 变换为

$$\Psi(\omega) = \pi^{-\frac{1}{4}}\left[e^{-\frac{(\omega-\omega_0)^2}{2}} - e^{-\frac{\omega_0^2}{2}} \cdot e^{-\frac{\omega^2}{2}}\right] \tag{3-238}$$

由上式可以看出它满足容许条件，即 $\omega=0$ 时 $\Psi(0)=0$。如果直接求取容许条件，可作如下运算：

$$\begin{aligned}
\int_{-\infty}^{+\infty}\psi(t)\mathrm{d}t &= \int_{-\infty}^{+\infty}\pi^{-\frac{1}{4}}(e^{-j\omega_0 t} - e^{-\frac{\omega_0^2}{2}})e^{-\frac{t^2}{2}}\mathrm{d}t \\
&= \pi^{-\frac{1}{4}}\left[\int_{-\infty}^{+\infty}e^{-j\omega_0 t}e^{-\frac{t^2}{2}}\mathrm{d}t - \int_{-\infty}^{+\infty}e^{-\frac{\omega_0^2}{2}}e^{-\frac{t^2}{2}}\mathrm{d}t\right] \\
&= \pi^{-\frac{1}{4}}\left[\sqrt{2\pi}e^{-\frac{\omega_0^2}{2}} - e^{-\frac{\omega_0^2}{2}}\int_{-\infty}^{+\infty}e^{-\frac{t^2}{2}}\mathrm{d}t\right] \\
&= \pi^{-\frac{1}{4}}\left[\sqrt{2\pi}e^{-\frac{\omega_0^2}{2}} - \sqrt{2\pi}e^{-\frac{\omega_0^2}{2}}\right] = 0
\end{aligned}$$

当 $\omega_0 \geqslant 5$ 时 $e^{-\frac{\omega_0^2}{2}} \approx 0$，所以，式(3-237)的第二项可以忽略，可近似表示为

$$\psi(t) = \pi^{-\frac{1}{4}}e^{-j\omega_0 t}e^{-\frac{t^2}{2}} \qquad \omega_0 \geqslant 5 \tag{3-239}$$

相应的 Fourier 变换为

$$\Psi(\omega) = \pi^{-\frac{1}{4}}e^{-\frac{(\omega-\omega_0)^2}{2}} \tag{3-240}$$

尺度为 a 的 Morlet 小波 $\psi_{a,0}(t)$ 的 Fourier 变换是

$$\Psi_{a,0}(\omega) = a\pi^{-\frac{1}{4}}e^{-(\omega_0-a\omega)^2/2} = a\pi^{-\frac{1}{4}}e^{-a^2(\frac{\omega_0}{a}-\omega)^2/2}$$

它的支撑区几乎是 $\omega>0$ 的整个区域，因为，在 $\omega<0$ 和 $\omega_0>5$ 时，$e^{-(\omega_0-a\omega)^2/2} < e^{-\frac{\omega_0^2}{2}} \approx 0$，所以，$\Psi(\omega)$ 可忽略不计。$\Psi_{a,0}(\omega)$ 的支撑区中心在 $\frac{\omega_0}{2}$ 处，波形宽度 $\sigma_{\Psi_{a,b}} = \frac{1}{a}$，并随着 a 的减小向外扩展。而小波 $\psi_{a,b}(t)$ 本身的中心在 b 处，以 $\sigma_{\psi_{a,b}} = a$ 的方式随 a 的增大向外扩展。Morlet 小波如图 3-27 所示。

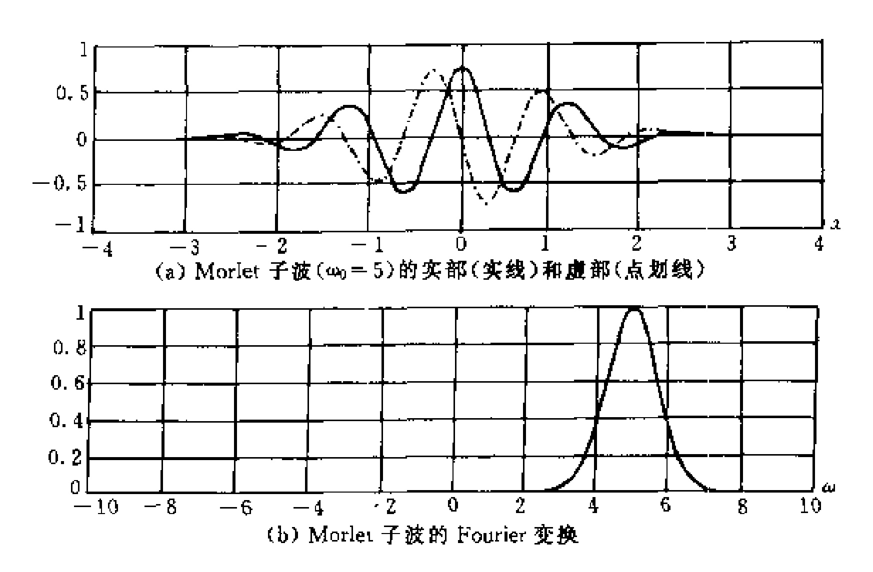


图 3-27 Morlet 小波及其 Fourier 变换

4. 小波变换的基本性质

(1) 线性

小波变换是线性变换。

设 $W_{f_1}(a,b)$ 为 $f_1(t)$ 的小波变换，

则有：

$$f(t)=\alpha f_1(t)+\beta f_2(t)$$
$$W_f(a,b)=\alpha W_{f_1}(a,b)+\beta W_{f_2}(a,b) \tag{3-241}$$

(2) 平移和伸缩的共变性

连续小波变换在任何平移之下是共变的，即

若 $f(t)\Leftrightarrow W_f(a,b)$ 是一对小波变换关系，则 $f(t-b_0)\Leftrightarrow W_f(a,b-b_0)$ 也是小波变换关系。

对于任何伸缩也是共变的，即：

若 $f(t)\Leftrightarrow W_f(a,b)$，则

$$f(a_0t)\Leftrightarrow \frac{1}{\sqrt{a_0}}W_f(a_0a,a_0b) \tag{3-242}$$

证明：
$$W_f(a,b)=\int_{-\infty}^{+\infty}f(a_0t)\psi\left(\frac{t-b}{a}\right)\mathrm{d}t$$

作变量替换 $a_0t=x$，$t=\frac{x}{a_0}$，$\mathrm{d}t=\frac{\mathrm{d}x}{a_0}$，代入上式，则有

$$W_f(a,b)=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty}f(x)\psi\left[\frac{\frac{x}{a_0}-b}{a}\right]\frac{1}{a_0}\mathrm{d}x$$
$$=\frac{1}{\sqrt{a_0}}\cdot\frac{1}{\sqrt{a_0a}}\int_{-\infty}^{+\infty}f(x)\psi\left(\frac{x-a_0b}{a_0a}\right)\mathrm{d}x=\frac{1}{\sqrt{a_0}}W_f(a_0a,a_0b)$$

(3) 微分运算

$$W_{\psi_{a,b}}\left(\frac{\partial^n f(t)}{\partial t^n}\right)=(-1)^n\int_{-\infty}^{+\infty}f(t)\left(\frac{\partial^n}{\partial t^n}\right)[\overline{\psi_{a,b}(t)}]\mathrm{d}t \tag{3-243}$$

(4) 局部正则性

如果函数在 t_0 处 n 阶连续可微，即 $f\in C^n(t_0)$

则有

$$W_f(a,b)\lesssim a^{n+1}a^{\frac{1}{2}} \tag{3-244}$$

说明函数的局部性质与函数小波局部性质有关。也就是说小波变换能够度量函数的局部正则性。

除上述性质外，小波变换还有诸如能量守恒性、空间-尺度局部化等特性。

3.6.4 离散小波变换

1. 离散小波的定义

在连续小波变换中，伸缩参数和平移参数连续取值，连续小波变换主要用于理论分析，在实际应用中离散小波变换更适于计算机处理。离散小波的定义可由下式表示：

$$\psi_{m,n}(t)=\frac{1}{\sqrt{a_0^m}}\psi\left(\frac{t-nb_0a_0^m}{a_0^m}\right)=a_0^{-\frac{m}{2}}\psi(a_0^{-m}t-nb_0) \tag{3-245}$$

相应的离散小波变换可由下式定义：

$$\langle f,\psi_{m,n}\rangle=a_0^{-\frac{m}{2}}\int_{-\infty}^{+\infty}f(t)\psi_{m,n}(t)\mathrm{d}t=a_0^{-\frac{m}{2}}\int_{-\infty}^{+\infty}f(t)\psi(a_0^{-m}t-nb_0)\mathrm{d}t \tag{3-246}$$

2. 正交小波变换

连续小波可以刻画函数 $f(t)$ 的性质和变化过程，用离散小波也可以刻画 $f(t)$。按调和分析方法，把 $f(t)$ 写成级数展开形式，就构成了 n 维空间中函数逼近问题。

在数学中，"空间"是用公理确定了元素之间的关系的集合，例如：距离空间是定义了元素间距离的集合；定义了元素范数的线性空间叫做线性赋范空间等，在离散小波变换中赋范空间和内积空间的概念是很重要的。

在 $[a,b]$ 上 p 次可积的函数空间 $L^p\left\{\int_a^b [f(t)]^p \mathrm{d}t < \infty\right\}$，我们定义范数为

$$\| f(t) \|_p = \left\{\int_a^b |f(t)|^p \mathrm{d}t\right\}^{\frac{1}{p}} \tag{3-247}$$

对于 p 次可和的数列空间 $l^p\left\{\sum_{n=1}^{+\infty} |x_n|^p < \infty\right\}$

定义范数为

$$\| x \| = \left\{\sum_{n=1}^{+\infty} |x_n|^p\right\}^{\frac{1}{p}} \tag{3-248}$$

完备的内积空间称做 Hilbert 空间，内积的定义如下：

线性函数空间 $L^2(a,b)$，

$$\langle x,y\rangle = \int_a^b x(t)y(t)\mathrm{d}t \qquad x,y \in L^2(a,b) \tag{3-249}$$

内积空间中的范数为

$$\| x \| = \langle x,x\rangle^{\frac{1}{2}} = \left\{\sum_{n=1}^{+\infty} |x_n|^2\right\}^{\frac{1}{2}} \tag{3-250}$$

由离散小波的定义，如果把 t 也离散化，并选择 $a_0=2, b_0=1$，则可得到二进小波：

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\psi\left(\frac{t-n2^m}{2^m}\right) = 2^{-\frac{m}{2}}\psi(2^{-m}t-n) \tag{3-251}$$

设 $\psi_{0,0}\equiv\psi(t)$ 可构造出正交小波 $\psi_{m,n}(t)$，

即：

$$\int \psi_{m,n}(t)\psi_{m',n'}(t)\mathrm{d}t = \begin{cases}1 & m=m' \quad n=n' \\ 0 & \text{其他}\end{cases} \tag{3-252}$$

所以

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\psi\left(\frac{t-n2^m}{2^m}\right) = 2^{-\frac{m}{2}}\psi(2^{-m}t-n) \tag{3-253}$$

这就是二进正交小波。由二进正交小波可得到信号 $f(t)$ 的任意精度的近似表示。由此：

$$f(t) = \sum_{m=-\infty}^{+\infty}\sum_{m=-\infty}^{+\infty}\langle f,\psi_{m,n}\rangle\psi_{m,n}(t) \tag{3-254}$$

3. 尺度函数

由尺度函数构造小波是小波变换的必径之路。尺度函数应 $\varphi(t)$ 满足下列条件：

1) $\int_{-\infty}^{+\infty}\varphi(t)\mathrm{d}t = 1$，它是一个平均函数，与小波函数 $\psi(t)$ 相比较，其傅里叶变换 $\Phi(\omega)$ 具有低通特性，$\Psi(\omega)$ 具有带通特性。

2) $\|\varphi(t)\|=1$，尺度函数是范数为 1 的规范化函数。

3) $\int_{-\infty}^{+\infty}\varphi_{m,n}(t)\psi_{m',n'}(t)\mathrm{d}t = 0$，即尺度函数对所有的小波是正交的。

4) $\int_{-\infty}^{+\infty}\varphi_{m,n}(t)\varphi_{m',n'}(t)\mathrm{d}t = 0$，即尺度函数对于平移是正交的，但对于伸缩 m 来说不是正交的。

5) $\varphi(t) = \sqrt{2}\sum_{n\in Z}h_n\varphi(2t-n)$，即某一尺度上的尺度函数可以由下一尺度的线性组合得到，h_n 是尺度系数。

$\varphi(t)$的 Fourier 变换为

$$\Phi(t) = \sum_{n\in Z}\frac{h_n}{\sqrt{2}}\Phi\left(\frac{\omega}{2}\right)\mathrm{e}^{-\frac{\mathrm{j}\omega n}{2}} = H\left(\frac{\omega}{2}\right)\cdot\Phi\left(\frac{\omega}{2}\right) \tag{3-255}$$

式中：

$$H\left(\frac{\omega}{2}\right) = \sum_{n\in Z}\frac{h_n}{\sqrt{2}}\mathrm{e}^{-\frac{\mathrm{j}\omega n}{2}} \tag{3-256}$$

6) 尺度函数与小波是有关连的。$\psi(t)$可表示如下：

$$\psi(t) = \sqrt{2}\sum_{n\in Z}g_n\varphi(2t-n) \tag{3-257}$$

式中 $\sqrt{2}$ 是归一化因子，g_n 是由尺度系数 h_n 导出的系数，相应的 Fourier 变换为

$$\Psi(\omega) = \sum_{n\in Z}\frac{g_n}{\sqrt{2}}\mathrm{e}^{-\frac{\mathrm{j}\omega n}{2}}\Phi\left(\frac{\omega}{2}\right) = G\left(\frac{\omega}{2}\right)\Phi\left(\frac{\omega}{2}\right) \tag{3-258}$$

这里：

$$G\left(\frac{\omega}{2}\right) = \sum_{n\in Z}\frac{g_n}{\sqrt{2}}\mathrm{e}^{-\frac{\mathrm{j}\omega n}{2}}$$

这说明小波可以由尺度函数的伸缩和平移的线性组合获得，这就是构造小波正交基的途径。

从上述关系可以看出，从 $\varphi(t)$导出 $\psi(t)$的关键在于建立 h_n 和 g_n 的正交关系。其中，

$$g_n = (-1)^{1-n}h_{1-n} \qquad n\in Z \tag{3-259}$$

$$h_n = \sqrt{2}\langle\varphi(t),\varphi(2t-n)\rangle = \sqrt{2}\int_{-\infty}^{+\infty}\varphi(t)\,\overline{\varphi(2t-n)}\mathrm{d}t \tag{3-260}$$

关于 $\psi(t)$的求解可用：

$$\Phi(\omega) = \sum_{n\in Z}\frac{h_n}{\sqrt{2}}\Phi\left(\frac{\omega}{2}\right)\mathrm{e}^{-\frac{\mathrm{j}\omega n}{2}} = H\left(\frac{\omega}{2}\right)\Phi\left(\frac{\omega}{2}\right)$$

反复迭代求得 $\Phi(\omega)$的极限形式：

$$\Phi(\omega) = \left[\prod_{i=1}^{k}H\left(\frac{\omega}{2^i}\right)\right]\Phi\left(\frac{\omega}{2^k}\right) \approx \prod_{i=1}^{\infty}H\left(\frac{\omega}{2^i}\right)\Phi(0) = \prod_{i=1}^{\infty}H\left(\frac{\omega}{2^i}\right) \tag{3-261}$$

然后求 Fourier 逆变换：

$$\varphi(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\prod_{i=1}^{\infty}H\left(\frac{\omega}{2^i}\right)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega \tag{3-262}$$

4. 紧支集概念

函数 $f(t)$的支集或支撑区 supp{f}是指最大开集 E 的补集。对于开集 E，$t\in E$ 有 $f(t)=0$，因此，函数的支集就是函数定义域的闭子集，也就是说，这样一个最小的闭子集或区间$[a,b]$，使得在$[a,b]$之外，函数 $f(t)$为零。如果说函数 $f(t)$是紧支集就是指 $f(t)$的支撑区 supp{f}是紧支集，即 supp{f}$\subset[a,b]$，$[a,b]$是有界闭区间。与紧支集概念相联系的是函数的平滑性

和速降性。

(1) 平滑性

如果$\frac{d^n f(t)}{dt^n}$对$\forall n\in N$是连续函数，则函数$f(t)$是平滑的。若$\frac{d^n f(t)}{dt^n}$对$\forall$ $(0\leqslant n\leqslant d)$是连续的，则函数$f(t)$的平滑度为$d$。平滑性决定了$\Phi(\omega)$的频率分辨率的高低。

(2) 速降性

函数$f(t)$对于$\forall n\in N$，存在一个有限常数$K_n>0$，使得$\forall t\in R$都有$|t^n f(t)|<K_n$，则说函数$f(t)$有无限速降性。速降性决定了$\varphi(t)$构造小波$\psi(t)$支集性质，也决定了其空间局部性的好坏与否。

通常我们希望小波是紧支撑的，因为这样的小波具有更好的局部特性，也有利于算法的实现，可惜的是紧支撑与平滑性二者不可兼得。

5. 非正交小波变换

构造一个既具有正交性，又具有紧支集、平滑性甚至对称性的小波基函数具有很大的困难。但在应用中，紧支集是保证优良的空间局部性的条件；对称性可使小波滤波特性具有线性相移特性，以避免信号失真；平滑性与频率分辨率有关。这些矛盾的实质是对小波基函数正交性的要求，如果对正交性不作苛求的话，即容许子波基函数有一定的相关性，可能会带来许多好处。由此，我们引入“框架”的概念。

如果存在两个称作框架界的常数A和B，且$0<A\leqslant B<\infty$，使得对于所有的Hilbert空间H中的函数$f(t)$满足下列关系：

$$A\|f\|^2\leqslant\sum_{l=-\infty}^{\infty}|\langle f,\varphi_l\rangle|^2\leqslant B\|f\|^2 \tag{3-263}$$

我们把Hilbert空间的一族函数$\{\varphi_l\in H: l\in Z\}$称作框架，其中常数$B<\infty$保证了变换$f\to\{\langle f,\varphi_l\rangle\}$是连续的；常数$A>0$保证了变换是可逆的，并有连续的逆变换。这样就可以用框架$\{\varphi_l\}$完全刻画函数$f(t)$，也可完全重构$f(t)$。一般情况框架不是正交基，它提供了对函数$f(t)$的一种冗余表示。这种表示使得恢复信号$f(t)$的数值计算十分稳定，而且对噪声也具有鲁棒性(Robustness)。

当$A=B$时的框架称之为紧框架(Tight frame)。此时，$f(t)$的简单展开式如下：

$$f(t)=\frac{1}{A}\sum_l\langle f,\varphi_l\rangle\varphi_l(t) \tag{3-264}$$

如果$A=B=1$，且$\|\varphi_l\|=1$则$\{\varphi_l\}$形成规范正交基，于是可得到通常的展开式。当$\{\psi_{m,n}\}$构成紧框架时，则有$A=B=C_\psi/b_0\lg a_0$。其中C_ψ是容许条件，即：

$$C_\psi=\int_{-\infty}^{+\infty}\frac{|\Psi(\omega)|^2}{\omega}d\omega<\infty \tag{3-265}$$

实际上，A严格等于B很困难，只能A接近B。即$\varepsilon=\frac{B}{A}-1\ll 1$，这种框架叫几乎紧框架(Snug Frame)，此时，展开式可由下式给出：

$$f(t)=\frac{2}{A+B}\sum_l\langle f,\varphi_l\rangle\varphi_l+\gamma \tag{3-266}$$

其中γ是误差。

令L表示一种映射关系，即$L: f(t)\to\{\langle f,\psi_{m,n}\rangle\}$

其中

$$\psi_{m,n}=\frac{1}{\sqrt{a_0^m}}\psi\left(\frac{t-nb_0a_0^m}{a_0^m}\right)=a_0^{-\frac{m}{2}}\psi(a_0^{-m}t-nb_0) \tag{3-267}$$

如果映射满足　　$A\|f\|^2\leqslant\sum_m\sum_n|\langle f,\psi_{m,n}\rangle|^2\leqslant B\|f\|^2$

则可通过小波系数$\{\langle f,\psi_{m,n}\rangle\}$刻画函数 $f(t)$。

如果$\{\psi_{m,n}\}$是一个紧框架，则有：

$$f(t)=\frac{1}{A}\leqslant\sum_m\sum_n\langle f,\psi_{m,n}\rangle\psi_{m,n}(t) \tag{3-268}$$

如果$\{\psi_{m,n}\}$是一个几乎紧框架，则：

$$f(t)=\frac{2}{A+B}\leqslant\sum_m\sum_n\langle f,\psi_{m,n}\rangle\psi_{m,n}+\gamma \tag{3-269}$$

只要 $\psi(t)$满足 $\int_{-\infty}^{+\infty}\psi(t)\mathrm{d}t=0$ 成立，且为紧支集或速降的，那么适当地选择 a_0，b_0 就可构造这样的框架。

Daubechies 给出了选择 a_0 和 b_0 的关系式：

$$A\leqslant\frac{\pi}{b_0\ln a_0}\int_{-\infty}^{+\infty}|\Psi(\omega)|^2|\omega|^{-1}\mathrm{d}\omega\leqslant B \tag{3-270}$$

其中 a_0，b_0 的选择条件很宽。例如 Mexico Hat 小波，当 $a_0=2$，$b_0=1$ 时，框架界 $A=3.223$，$B=3.596$，$\frac{B}{A}=1.116$。

6．关于 Daubechies 小波

由多分辨分析得到的尺度函数 $\varphi(t)$与小波函数 $\psi(t)$的双尺度差分方程为

$$\varphi(t)=\sqrt{2}\sum_{n=0}^{2N-1}h_n\varphi(2t-n) \tag{3-271}$$

$$\psi(t)=\sqrt{2}\sum_{n=0}^{2N-1}g_n\varphi(2t-n) \tag{3-272}$$

用尺度函数 $\varphi(t)$构造的小波 $\psi(t)$可通过伸缩和平移形成正交集，而且 $\psi(t)$具有紧支集、平滑性、对称性是十分困难的。为此 Daubechies 提出了一种解决办法。她的思路是先由 $\varphi(t)=\sqrt{2}\sum_{n=0}^{2N-1}h_n\varphi(2t-n)$ 的 Fourier 变换：

$$\Phi(2\omega)=H(\omega)\Phi(\omega)\qquad H(\omega)=\sum_n h_n\mathrm{e}^{-\mathrm{j}n\omega} \tag{3-273}$$

求出 $H(\omega)$再通过无穷乘积来定义：

$$\Phi(\omega)=\prod_{j=1}^{+\infty}H(2^{-j}\omega)$$

讨论使$\{\varphi(t-n)\}_{n\in Z}$的标准正交条件，为最后构造小波创造了条件。

设 $H(\omega)=\sum_n h_n\mathrm{e}^{-\mathrm{j}n\omega}$ 是三角多项式，系数$\{h_n\}$是实数，则有

$$H(\omega)=\left[\frac{1}{2}(1+\mathrm{e}^{\mathrm{j}\omega})\right]^N Q(\mathrm{e}^{-\mathrm{j}\omega})\qquad N\in Z. \tag{3-274}$$

Q 是实系数代数多项式，因$\overline{Q(\mathrm{e}^{\mathrm{j}\omega})}=Q(\mathrm{e}^{\mathrm{j}\omega})$，所以$|Q(\mathrm{e}^{-\mathrm{j}\omega})|^2=Q(\mathrm{e}^{\mathrm{j}\omega})Q(\mathrm{e}^{-\mathrm{j}\omega})$是 ω 的偶函数。将其表示成 $\cos\omega$ 的多项式，利用$\frac{(1-\cos\omega)}{2}=\sin^2\left(\frac{\omega}{2}\right)$，可等价地表示为 $\sin^2\left(\frac{\omega}{2}\right)$的多项式，记

为 $P(y)$，$y=\sin^2\left(\frac{\omega}{2}\right)$，注意到 $\left|\frac{(1+e^{j\omega})}{2}\right|^2=\left|\cos^2\frac{\omega}{2}\right|$，则有

$$\begin{aligned}|H(\omega)|^2&=\left[\cos^2\left(\frac{\omega}{2}\right)\right]^N|Q(e^{-j\omega})|^2\\&=\left[1-\sin^2\left(\frac{\omega}{2}\right)\right]^N|Q(e^{-j\omega})|^2\\&=(1-y)^2P(y)\end{aligned}\tag{3-275}$$

而

$$\begin{aligned}|H(\omega+\pi)|^2&=\left[\sin^2\left(\frac{\omega}{2}\right)\right]^N|Q(e^{-j(\omega+\pi)})|^2\\&=\left[\sin^2\left(\frac{\omega}{2}\right)\right]^N P[\sin^2(\omega+\pi)/2]\\&=y^N P\left[\cos^2\left(\frac{\omega}{2}\right)\right]\\&=y^N P\left[1-\sin^2\left(\frac{\omega}{2}\right)\right]=y^N P(1-y)\end{aligned}\tag{3-276}$$

因此，

$$|H(\omega)|^2+|H(\omega+\pi)|^2=(1-y)^N P(y)+y^N P(1-y)=1\tag{3-277}$$

$$P(y)\geqslant 0\qquad y\in[0,1]$$

由 Riesz 定理，求出 $|Q(e^{j\omega})|^2$。

$$|Q(e^{j\omega})|^2=\sum_{K=0}^{N-1}C_{N+K-1}^{K}\left(\sin^2\frac{\omega}{2}\right)^K+\left(\sin^2\frac{\omega}{2}\right)^N\cdot R\left(\frac{1}{2}-\sin^2\frac{\omega}{2}\right)$$

其中 $R(x)=R(-x)$，$R\left(\frac{1}{2}-\sin^2\frac{\omega}{2}\right)$ 可写成 $R\left(\frac{1}{2}\cos\omega\right)$，在最简单的情况下 $R(x)=0$，上式可简化为

$$|Q(e^{j\omega})|^2=\sum_{K=0}^{N-1}C_{N+K-1}^{K}\sin^{2K}\frac{\omega}{2}\tag{3-278}$$

求出 $|Q(e^{j\omega})|$，代入

$$H(\omega)=\left[\frac{1}{2}(1+e^{j\omega})\right]^N Q(e^{-j\omega})\tag{3-279}$$

就可以确定 $H(\omega)$ 及尺度系数 h_n 及小波系数 g_n。这样由 $H(\omega)$ 和 Riesz 定理给出的小波基函数称做 Danbechies 紧支集小波。对应的尺度函数和小波函数为

$$N\varphi(t)=\sqrt{2}\sum_{n=0}^{2N-1}h_n\varphi(2t-n)\tag{3-280}$$

$$N\psi(t)=\sqrt{2}\sum_{n=0}^{2N-1}g_n\varphi(2t-n)\tag{3-281}$$

$$g_n=(-1)^n h_{2N-n-1}\qquad n=0,1,2,\cdots,2N-1$$

7. B 样条小波分析

样条是一类分段光滑又在各段连接处具有一定光滑性的函数。它在数据的插值、拟合与平滑方面有很好的稳定性和收敛性，是函数逼近的有力工具。样条函数可以表示成变量 t 的多项式，在小波分析中用得最多的是 B 样条函数(Cardinal B-spline)。B 样条具有最小的支撑长度，而且有利于计算机实时处理。

这里把分段常数空间记为 S_1，分段多相式空间记为 S_m，m 是多项式的阶数。当 m 为正整

数时 S_m 称为基数样条空间，这是样条小波的基本空间。

m 阶 B 样条是 Haar 尺度函数与其自身作 m 次卷积运算后所得的函数，记为 $N_m(t)$，可得到：

$$N_1(t) = \begin{cases} 1 & 0 \leqslant t < 1 \\ 0 & \text{其他} \end{cases} \tag{3-282}$$

$$N_2(t) = N_1 * N_1(t) = \int_0^1 N_1(\tau) N_1(t-\tau)\mathrm{d}\tau = \begin{cases} t & 0 \leqslant t < 1 \\ 2-t & 1 \leqslant t < 2 \\ 0 & \text{其他} \end{cases} \tag{3-283}$$

$$N_3(t) = N_2 * N_1(t) = \int_0^1 N_2(t-\tau)\mathrm{d}\tau = tN_1(t) - (2-t)N_2(t)$$

$$= \begin{cases} \dfrac{1}{2}t^2 & 0 \leqslant t < 1 \\ \dfrac{3}{4} - \left(t - \dfrac{3}{2}\right)^2 & 1 \leqslant t < 2 \\ \dfrac{1}{2}(t-3)^2 & 2 \leqslant t < 3 \\ 0 & \text{其他} \end{cases} \tag{3-284}$$

即：

$$\begin{aligned} N_m(t) &= N_{m-1} * N_1(t) = N_{m-2} * N_1(t) * N_1(t) \\ &= \cdots = N_1 * N_1 * \cdots * N_1(t) \end{aligned} \tag{3-285}$$

其中 $N_1(t)$就是定义域[0,1)上为常数的特征函数 $\chi_{[0,1)}(t)$，B 样条有如下递推公式：

$$N_m(t) = \frac{t}{m-1}N_{m-1}(t) + \frac{m-t}{m-1}N_{m-1}(t-1) \tag{3-286}$$

图 3-28 示出了 $N_1(t)$，$N_2(t)$，$N_3(t)$的波形，由图可见 $N_1(t)$不连续，$N_2(t)$连续，但一阶导数不连续，$N_3(t)$有连续的一阶导数，因此，它比较常用。

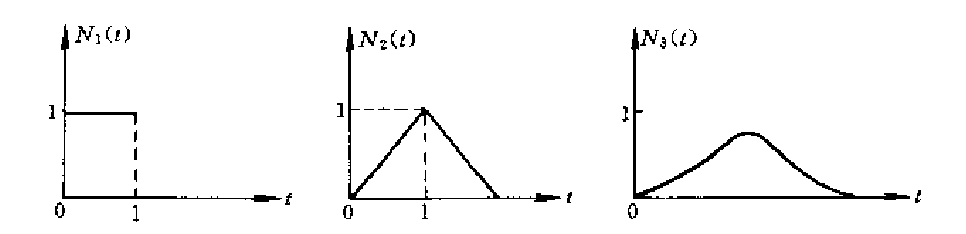


图 3-28 $m=1,2,3$ 时的基数 B 样条波形

如果基数 B 样条函数表示为对称于原点垂直轴的形式，则相应的波形如图 3-29 所示。

$N_m(t)$在频域中的形式可由 Fourier 变换得到：

$$\hat{N}_m(\omega) = \left(\frac{1-\mathrm{e}^{-\mathrm{j}\omega}}{\mathrm{j}\omega}\right)^m \tag{3-287}$$

当 $m=1$ 时就是 Haar 系，当 $m=2$ 时就是 Franklin 小波。在一般情况下，$N_m(t) \in C^m$。其支撑区是$[0,m]$，其正交尺度函数表达式为

$$\Phi_m(\omega) = \left(\frac{\sin \omega/2}{\omega/2}\right)^m \left[P\cos\frac{\omega}{2}\right]^{\frac{1}{2}} \tag{3-288}$$

这里，P 是 $2m$ 阶多项式，当 m 是偶数时，该展开式为

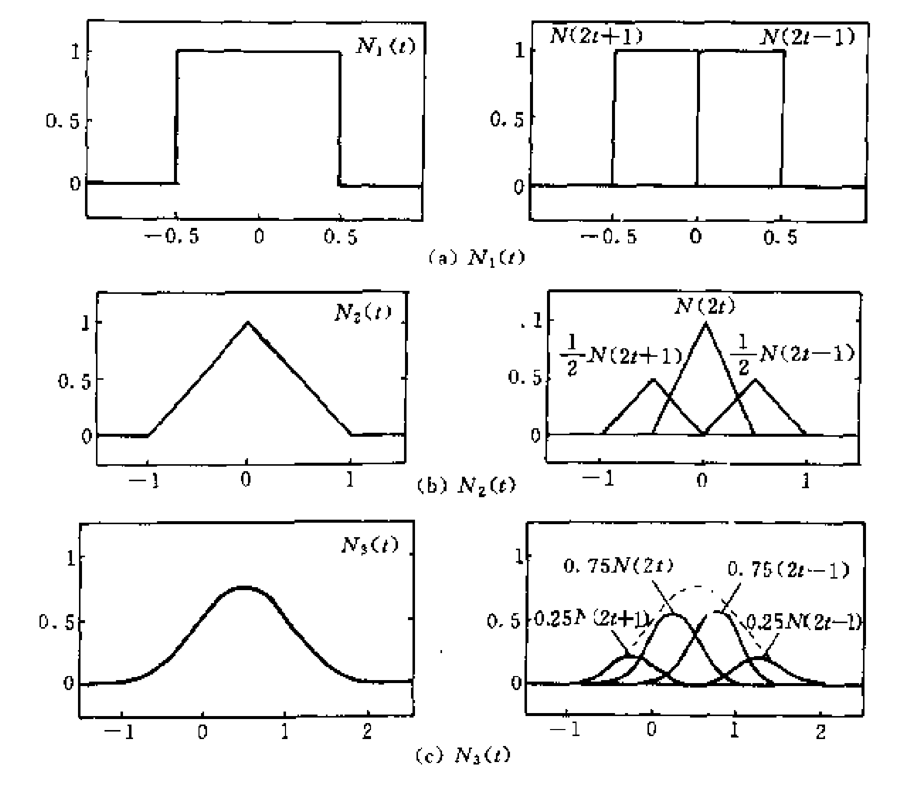


图 3-29 B样条函数的对称形式与构造

$$\Phi_m(\omega) = \left(\frac{\sin \omega/2}{\omega/2}\right)^m e^{-\frac{j\omega}{2}}\left[P\left(\cos\frac{\omega}{2}\right)\right]^{-\frac{1}{2}} \tag{3-289}$$

经双尺度方程求出 $\psi_m(t)$：

$$\psi_m(t) = \sqrt{2}\sum_k(-1)^k h_{1-k}\varphi_m(2t-k) \tag{3-290}$$

又可通过 $\Phi_m(\omega)$的 Fourier 反变换求出 $\varphi_m(t)$。

研究基数样条函数的目的是为了构造具有紧支撑的样条小波。首要工作是构造尺度函数 $\varphi_m(t)$．对于基数样条函数来说，其简单的构造公式如下：

$$\Phi_m(\omega) = \frac{\hat{N}_m(\omega)}{\left[\sum_{k=-\infty}^{+\infty}|\hat{N}_2(\omega+2k\pi)|^2\right]^{\frac{1}{2}}} \tag{3-291}$$

其中：

$$\sum_{k=-\infty}^{+\infty}|\hat{N}_m(\omega+2k\pi)|^2 = \frac{-\sin^{2m}t}{(2m-1)!}\cdot\frac{d^{2m-1}}{dt^{2m-1}}\cot t$$

当 $m=1$ 时，$\sum_{k=-\infty}^{+\infty}|\hat{N}_1(\omega+2k\pi)|^2 = 1$

当 $m=2$ 时，$\sum_{k=-\infty}^{+\infty}|\hat{N}_2(\omega+2k\pi)|^2 = \frac{1}{3}+\frac{2}{3}\cos^2\frac{\omega}{2}$

求得 $\Phi_m(\omega)$后即可构造规范正交系$\{\varphi(t-n)\}_{n\in Z}$，对于 $m=2$，$\Phi_2(\omega)$则有如下解析表达式：

$$\Phi_2(\omega) = \frac{\hat{N}_2(\omega)}{\left[\sum_{k=-\infty}^{+\infty} |\hat{N}_2(\omega + 2k\pi)|^2\right]^{\frac{1}{2}}} = \frac{e^{-j\omega}\left(\sin\frac{\omega}{2} / \frac{\omega}{2}\right)}{\left(\frac{1}{3} + \frac{2}{3}\cos^2\frac{\omega}{2}\right)^{\frac{1}{2}}} \tag{3-292}$$

然后在频域或时域中推导样条小波 $\psi_m(t)$，最简单的方法是由 $\Phi_m(\omega) = H_m\left(\frac{\omega}{2}\right)\Phi_m\left(\frac{\omega}{2}\right)$ 求出 $H_m\left(\frac{\omega}{2}\right)$，再由下式求出 $\Psi_m(\omega)$：

$$\Psi_m(\omega) = e^{-\frac{j\omega}{2}}\overline{H\left(\frac{\omega}{2} + \pi\right)}\Phi_m\left(\frac{\omega}{2}\right) \tag{3-293}$$

此后，对其施以 Fourier 反变换就可以得到小波函数 $\psi_m(t)$。

当 $m=2$ 时，则滤波函数 $H_2(\omega)$ 可由下式求得：

$$H_2(\omega) = \frac{\Phi_2(2\omega)}{\Phi_2(\omega)} = \left(\frac{1 + \cos\omega}{2}\right)\left(\frac{2 + \cos\omega}{1 + 2\cos^2\omega}\right)^{\frac{1}{2}} \tag{3-294}$$

$$\Psi_2(\omega) = \frac{-\frac{16e^{\frac{-j\omega}{2}}}{\omega^2}\sin^4\frac{\omega}{4}\left(1 + 2\sin^2\frac{\omega}{2}\right)^{\frac{1}{2}}}{\left[\left(\frac{1}{3} - \frac{2}{3}\sin^2\frac{\omega}{4}\right)\left(3 - 8\sin^2\frac{\omega}{4} + 8\sin^4\frac{\omega}{4}\right)\right]^{\frac{1}{2}}} \tag{3-295}$$

同理可求得 $m=3$ 的 $\Phi_3(\omega)$ 和 $\psi_3(\omega)$。图 3-30 示出了 B 样条小波的基函数 $\varphi(t)$ 与小波函数 $\psi(t)$，图(a)是线性样条基函数，图(b)是二次样条基函数。

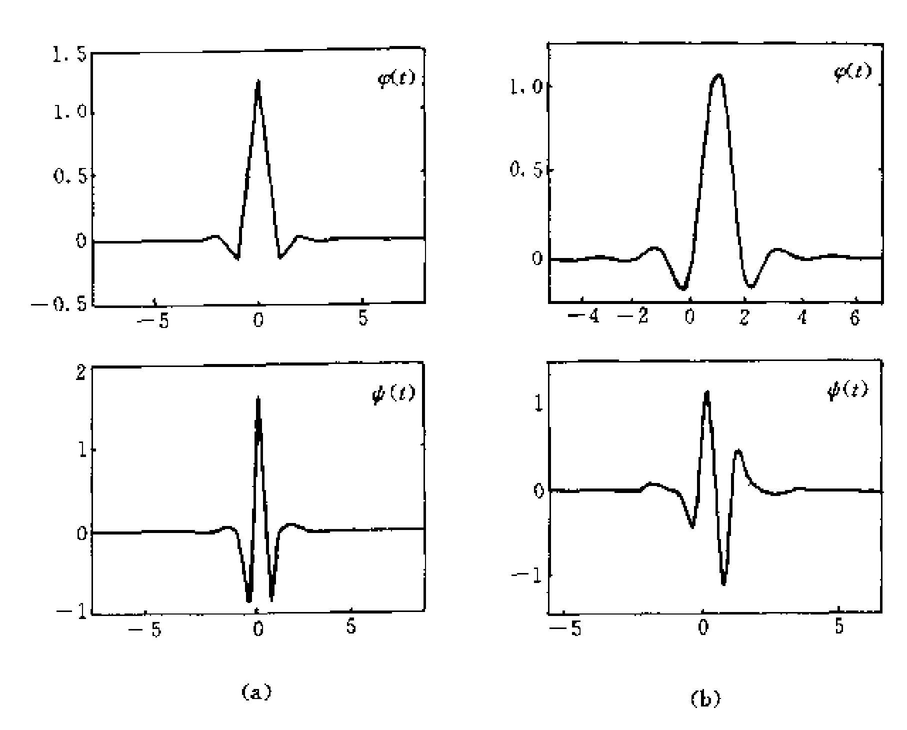


图 3-30　基数 B 样条小波波形

m 阶基数 B 样条的双尺度方程如下式所示：

$$\varphi_m(t) \equiv N_m(t) = \sum_{k=0}^{m} 2^{-(m-1)}\binom{m}{k} N_m(2t - k) \tag{3-296}$$

具有紧支撑的基数 B 样条小波 $\psi_m(t)$ 的表达式为

$$
\begin{aligned}
\psi_m(t) &= 2^{(m-1)} \sum_{k=0}^{2m-2} (-1)^k N_{2m}(k+1) N_{2m}^{(m)}(2t-k) \\
&= 2^{(m-1)} \sum_{k=0}^{3m-2} (-1)^k \left[\sum_{l=0}^{m} \binom{m}{l} N_{2m}(k-l+1) \right] N_m(2t-k)
\end{aligned}
\tag{3-297}
$$

式中：$N_{2m}^{(m)}(2t-k)$表示 $N_{2m}(2t-k)$的 m 阶导数。

样条小波是框架理论的一部分，基数 B 样条小波 $\psi_m(t)$和$\overline{\psi}_m$ 在 m 是偶数时是对称的，当 m 是奇数时是反对称的，它们都有广义线性相位。基数 B 样条小波是一个新的研究课题，在信号分析与图像压缩中得到较好的成果。

3.6.5 小波包

任一函数可以表示为小波展开，但小波函数 $\psi(t)$并不是惟一的，由于研究对象是多种多样的，究竟选择哪一种小波作为分解和重构的基函数是学者们关注的问题。因此，我们希望针对不同的处理信号能有一个选择基函数的准则。正像 Meyer 在 1990 年日本东京国际数学大会上指出的那样，小波分析固然是研究突变信号的有力工具，但在处理渐变信号时却不如 Gabor 分析，而在实际处理中两种信号总是交替出现的。因此，人们往往交替使用小波分析和窗口 Fourier 分析。正交小波包是一种建立选择"最好基"准则，并给出具体运算方法的数学工具，它对小波分析与综合应用是至关重要的。

一般来说，小波包分析包括小波基包和小波框架包，通俗地说就是从多分辨分析出发采用滤波的思路建立小波基库。在数据压缩方面，Coifman 和 Meyer 等人建立了一个广泛的函数目录库，称为小波包(Wavelet Packet)。由此构成了一个可数的无穷多正交基，该小波包将 Gabor 函数和小波函数统一为一个集，这个集通过尺度参数(频率参数)q，空间参数 k，振荡参数 n 控制零平均的局部化振荡函数，其中 k 对应中心位置，q 对应空间支撑宽度，n 对应空间振荡次数，于是通过一个"母小波"的伸缩和平移就可产生一个小波包族，对于给定的信号可选择最合适的函数来分解它，选择的准则可以是信息熵最小或其他。

小波包的基本数学模型可作如下分析：

令 $w_n(t)$满足双尺度方程，即

$$w_{2n}(t) = \sqrt{2} \sum_k h_k w_n(2t-k) \tag{3-298}$$

$$w_{2n+1}(t) = \sqrt{2} \sum_k g_k w_n(2t-k) \tag{3-299}$$

其中 h_n 与 g_n 间存在正交关系。

显然，$w_n(t)$当 $n=0$ 时就是尺度函数 $\varphi(t)$；当 $n=1$ 时就是小波函数 $\psi(t)$，

$$w_0(t) = \sqrt{2} \sum_k h_k w_0(2t-k) \tag{3-300}$$

$$w_1(t) = \sqrt{2} \sum_k g_k w_0(2t-k) \tag{3-301}$$

这里：$w_0(t)$为尺度函数 $\varphi(t)$，$w_1(t)$为小波函数 $\psi(t)$。

所以，函数集$\{w_n(t)\}$可以看成是 $w_1(t)=\psi(t)$的推广，用来统一表征尺度函数 $\varphi(t)$与小波函数 $\psi(t)$。如果 $w_{2n}(t)$对应尺度函数方程，$w_{2n+1}(t)$对应在小波函数方程，并引入下列符号：

$$U_j^0 = V_j \qquad j \in Z \tag{3-302}$$

$$V_j^1 = W_j \qquad j \in Z \tag{3-303}$$

这样，Hilbert 空间的正交分解 $V_{j+1}=V_j \oplus W_j$ 可用 U_{j+1}分解统一表示，即

$$U_{j+1}^0 = U_j^0 \oplus U_j^1 \qquad j \in Z \tag{3-304}$$

推广到一般情形，

$$U_{j+1}^n = U_j^{2n} \oplus U_j^{2n+1} \qquad j \in Z \tag{3-305}$$

这样，把 n 分解为 $2n$ 与 $2n+1$ 两部分，而 U_j^{2n} 和 U_j^{2n+1} 是 U_{j+1}^n 的子空间，U_j^{2n} 对应于 w_{2n}，U_j^{2n+1} 对应于 w_{2n+1}，可以用 W_{j+1}^n 记 U_{j+1}^n，得到小波空间 W_{j-1}^n 的分解：

$$W_{j+1}^n = U_{j-1}^n = U_j^{2n} \oplus U_j^{2n+1} \qquad j \in Z \tag{3-306}$$

由 $L^2(R) = \bigoplus_{j\in Z} W_j$ 可知，多分辨分析按不同的尺度 j 把 Hibert 空间 $L^2(R)$ 分解为子空间 $\{W_j\}_{j\in Z}$，针对上式(3-306)令 $n=1,2,\cdots$，$j=1,2,\cdots$；反复迭代可得到小波包如下：

$$\begin{aligned}
W_j &= U_{j-1}^2 \oplus U_{j-1}^3 \\
W_j &= U_{j-2}^4 \oplus U_{j-2}^5 \oplus U_{j-2}^6 \oplus U_{j-2}^7 \\
&\cdots \\
W_j &= U_{j-k}^{2^k} \oplus U_{j-k}^{2^k+1} \oplus \cdots \oplus U_{j-k}^{2^{k+1}-1} \\
&\cdots \\
W_j &= U_0^{2^j} \oplus U_0^{2^j+1} \oplus \cdots \oplus U_0^{2^{j+1}-1}
\end{aligned} \tag{3-307}$$

这种 W_j 空间分解的任意子空间序列表达式为

$$U_{j-k}^{2^k+m}, \qquad m = 0,1,2,\cdots,2^k-1,\ k=1,2,\cdots,\ j=1,2,\cdots$$

与此相对应的是规范正交基：

$$\{2^{\frac{(j-k)}{2}} w_{2^k+m}(2^{j-k}t - l);\ l \in Z\} \tag{3-308}$$

通常把 w_n 或 $\{2^{\frac{(j-k)}{2}} w_{2^k-m}(2^{j-k}t-l)\}$ 叫做小波包。

显然，尺度函数 $\varphi(t)$ 和小波函数 $\psi(t)$ 都是它的最简单形式。小波包与小波函数相比较它具有划分较高频率倍频程的能力，从而提高了频率的分辨率，能获得更好的频域局部化。

由上面的分析可见，对于小波 $\psi_{j,k}(t)$ 来说，按尺度 j 的二进分频方式，在 W_j 子空间的第 j 个频带内是提取局部信息的频率窗口，

$$H_j \equiv (2^{j-1}\sigma_\Psi, 2^{j+2}\sigma_\Psi) \tag{3-309}$$

σ_ψ 是小波的均方根带宽。

对于小波包 $U_{j-k}^{2^k+m}$ 来说，是把第 j 个频带 H_j 进一步二进划分方式细致分割为 2^k 个"子频带"$H_j^{k,m}$，$m=1,2,\cdots,2^k-1$，以便获得子频带内的局部化信息，尺度为 j 时，所有子频带 $H_j^{k,m}$ 的并就是整个第 j 个频带 H_j，即

$$\bigcup_{m=0}^{2^k-1} H_j^{k,m} = H_j \qquad k = 1,2,\cdots,j \tag{3-310}$$

成立。

这意味着把 Hilbert 空间 $L^2(R)$ 正交分解为 W 子空间和 U 子空间两部分，即

$$L^2(R) = \bigoplus_{j\in Z} W_j = \cdots \oplus W_{-2} \oplus W_{-1} \oplus W_1 \oplus U_0^2 \oplus U_0^3 \cdots \oplus \tag{3-311}$$

容易看出，小波包分解比小波分解增加了 $U_0^2, U_0^3, \cdots$ 成份。由 $\psi_{j,k}(k) = 2^{\frac{j}{2}}\psi(2^jt-k)$ 说明小波基函数涉及到尺度参数 j 和平移参数 k，也就是小波基函数由这两个参数来刻划，而小波包涉及 3 个参数，即：尺度参数 j，平移参数 k 和频率参数 f，小波包基一般表示为 $2^{\frac{j}{2}}\psi_f(2^jt-k)$。

研究小波包的目的在于建立小波包基库，以便从中选择最合适的基来分解信号或逼近被分析函数。这里便有一个选择准则问题，利用熵的概念，设 H 是 Hilbert 空间，$\gamma \in H$，且

$\| v \| = 1$，如果假设 $H=\bigoplus_j H_j$ 是空间 H 的正交直和，则有熵的定义为

$$E^2(v,\{H_j\}) = -\sum_i \| v_i \|^2 \ln \| v_i \|^2 \tag{3-312}$$

用熵作判据的理由是正交基的可加性可以用熵的可加性度量。

假定 E 预先给定，$x=\{x_i\}$ 是可分空间 V 中的数据序列或矢量，若从小波包基库中选出某一正交基为 B，而 B_x 表示以基 B 展开 x 时的系数序列 $\langle x,B\rangle$，如果 $E(B_x)$ 是最小的，则 B 是熵值最小意义下的最优基。B 选择算法比较简单，它是一种搜索算法。搜索步骤可按下式进行：

$$A_{k-1,n} = \begin{cases} A_{k,2n} \oplus A_{k,2n+1} & M_A < M_B \\ B_{k-1,n} & 其他 \end{cases} \tag{3-313}$$

这里：B_{kn} 表示按二进间隔划分的标准正交基，对于某一 k 值，假如对所有的 $0 \leqslant n < 2^k$ 我们选取 A_{kn}；对于 $0 \leqslant n < 2^{k-1}$ 选取 A_{k-1}，令熵 E 最小。其中，$M_B \equiv M(B^*_{k-1,n}x)$，$M_A \equiv M(A^*_{k,2n}x) + M(A^*_{k,2n+1}x)$ 分别表示 x 以基 $B^*_{k-1,n}$ 展开和以基 $(A^*_{k,2n}, A^*_{k,2n+1})$ 展开的熵值。

3.6.6 二维小波

由于图像和计算机视觉信息一般是二维或多维信息，因此，小波理论向二维或多维推广是十分重要的研究课题。目前，高维小波理论还远不如一维小波理论那样完善，而且，高维紧支集小波的构造也还没有形成通用的方法，所以，我们从应用出发，主要讨论二维小波。

1. 二维连续小波

(1) 二维连续小波变换的定义

二维连续小波以变换的定义如下：

$$\begin{aligned} \langle f,\psi_{a,b}\rangle &\equiv W_f(a,b_1,b_2) \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(t_1,t_2)\psi_{a,b}(t_1,t_2)\mathrm{d}t_1\mathrm{d}t_2 \qquad a>0 \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(t_1,t_2)\,\frac{1}{a}\psi_{a,b}\left(\frac{(t_1,t_2)-(b_1,b_2)}{a}\right)\mathrm{d}t_1\mathrm{d}t_2 \end{aligned} \tag{3-314}$$

逆变换为

$$f(t_1,t_2) = \frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{a=0}^{+\infty} a^{-3}W_f(a,b_1,b_2)\psi_{a,b}(t_1,t_2)\mathrm{d}a\mathrm{d}b_1\mathrm{d}b_2 \tag{3-315}$$

(2) 二维小波的容许条件

1)紧支撑集；

2)均值为零，即 $\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\psi(t_1,t_2)\mathrm{d}t_1\mathrm{d}t_2 = 0$

(3) 二维 Morlet 小波

在二维平面上定义矢量 $t=(t_1,t_2)$，且 $|t|=\sqrt{t_1^2+t_2^2}$，则 Morlet 小波定义为

$$\psi^\theta(t) = \frac{1}{\sqrt{\pi}}e^{-j\Omega^0 t}e^{\frac{|t|^2}{2}} \qquad |\Omega^0| \geqslant 5 \tag{3-316}$$

其 Fourier 变换为

$$\psi^\theta(\Omega) = \frac{1}{\sqrt{\pi}}e^{\frac{-|\Omega-\Omega^0|^2}{2}} \tag{3-317}$$

式中 $\Omega=(\omega_1,\omega_2)$，$\Omega^0=(\omega_1^0,\omega_2^0)$ 为一常数，上角标 θ 代表小波的方向，

$$\theta = \arctan\frac{\omega_2^0}{\omega_1^0} \tag{3-318}$$

2. 二维离散正交小波

由一维 $L^2(R)$ 正交小波基推广到二维 $L^2(R^2)$ 是很自然的思路，这种推广有三种不同的方法。

(1) 由尺度函数 $\psi(t)$ 出发建立多分辨分析

定义下式为二维尺度函数：

$$\Phi(t_1,t_2) = \sum_{n_1,n_2} h_{n_1,n_2}\Phi(2t_1 - n_1, 2t_2 - n_2) \qquad n_1,n_2 \in Z^2 \tag{3-319}$$

则有滤波函数为

$$H_0(\omega_1,\omega_2) = \frac{1}{2}\sum_{n_1,n_2} h_{n_1,n_2} e^{-j(n_1\omega_1 + n_2\omega_2)} \tag{3-320}$$

该滤波函数应满足正交条件：

$$|H_0(\omega_1,\omega_2)|^2 + |H_0(\omega_1+\pi,\omega_2)|^2 + |H_0(\omega_1,\omega_2+\pi)|^2 + |H_0(\omega_1+\pi,\omega_2+\pi)|^2 = 1 \tag{3-321}$$

然后求在相应的尺度函数 $\Phi(t_1,t_2)$ 下的正交小波基：

$$\Psi^\lambda(\omega_1,\omega_2)| = H_\lambda\left(\frac{\omega_1}{2} + \frac{\omega_2}{2}\right)\hat{\Phi}\left(\frac{\omega_1}{2} + \frac{\omega_2}{2}\right) \qquad \lambda = 1,2,3$$

当 $\lambda=1,2,3$ 时，

$$H_\lambda\left(\frac{\omega_1}{2} + \frac{\omega_2}{2}\right) = \begin{bmatrix} H_0(\omega_1,\omega_2) & H_1(\omega_1,\omega_2) & H_2(\omega_1,\omega_2) & H_3(\omega_1,\omega_2) \\ H_0(\omega_1+\pi,\omega_2) & H_1(\omega_1+\pi,\omega_2) & H_2(\omega_1+\pi,\omega_2) & H_3(\omega_1+\pi,\omega_2) \\ H_0(\omega_1,\omega_2+\pi) & H_1(\omega_1,\omega_2+\pi) & H_2(\omega_1,\omega_2+\pi) & H_3(\omega_1,\omega_2+\pi) \\ H_0(\omega_1+\pi,\omega_2+\pi) & H_1(\omega_1+\pi,\omega_2+\pi) & H_2(\omega_1+\pi,\omega_2+\pi) & H_3(\omega_1+\pi,\omega_2+\pi) \end{bmatrix} \tag{3-322}$$

(2) 由一维小波 $\psi(t)$ 出发定义二维正交小波

$$\psi_{m_1,n_1,m_2,n_2}(t_1,t_2) = \psi_{m_1,n_1}(t_1)\psi_{m_2,n_2}(t_2) \qquad (m_1,m_2,n_1,n_2) \in Z \tag{3-323}$$

(3) 由一维多分辨率分析出发

设有空间 V_m，$m\in Z$，引入一维多分辨分析张量积 $\otimes$，

$$V_0 = V_0 \otimes V_0 = \overline{\text{span}\{F(t_1,t_2) = f(t_1)y(t_2); f, y \in V_0\}}$$
$$F \in V_m \Leftrightarrow F(2^{-m} \cdot 2^{-m}) \in V_0 \otimes V_0 \tag{3-324}$$

V_m 应满足如下条件：

$$\cdots \boldsymbol{V}_2 \subset \boldsymbol{V}_1 \subset \boldsymbol{V}_0 \subset \boldsymbol{V}_{-1} \subset \boldsymbol{V}_{-2} \cdots$$
$$\bigcap_{m\in Z} \boldsymbol{V}_m = \{0\} \qquad \overline{\bigcup_{m\in Z} \boldsymbol{V}_m} = L^2(R^2) \tag{3-325}$$

二维尺度函数可定义为

$$\Phi_{0,n_1,n_2}(t_1,t_2) = \varphi(t_1, -n_1)\varphi(t_2 - n_2) \qquad n_1,n_2 \in Z \tag{3-326}$$

它是 $\boldsymbol{V}_0$ 的正交基，$\boldsymbol{V}_0$ 是由函数 Φ 的 Z^2 平移生成的。

对于尺度 $m\neq 0$ 的情况下，

$$\begin{aligned}\Phi_{m,n_1,n_2}(t_1,t_2) &= \varphi_{m,n_1}(t_1)\varphi_{m,n_2}(t_2) \\ &= 2^{-m}\varphi(2^{-m}t_1 - n_1)\varphi(2^{-m}t_2 - n_2)\end{aligned}$$

$$=2^{-m}\Phi(2^{-m}t_1-n_1,2^{-m}t_2-n_2)\qquad n_1,n_2\in Z \tag{3-327}$$

由此，可生成空间 V_m，如果用 W_m 表示 V_{m+1} 中 V_m 的正交补空间，则有

$$\begin{aligned}V_{m-1}&=V_{m-1}\otimes V_{m-1}=(V_m\oplus W_m)\otimes(V_m\oplus W_m)\\&=(V_m\otimes V_m)\oplus[(W_m\otimes V_m)\oplus(V_m\otimes W_m)\oplus(W_m\otimes W_m)]\\&=V_m\oplus W_m\end{aligned} \tag{3-328}$$

正交补空间 W_m 由三个子空间的直和组成，其中 $(W_m\otimes V_m)$ 由正交基 $\psi_{m,n_1}(t_1)\varphi_{m,n_2}(t_2)$ 生成，$(V_m\otimes W_m)$ 由 $\varphi_{m,n_1}(t_1)\psi_{m,n_2}(t_2)$ 给出，而 $(W_m\otimes W_m)$ 则对应 $\psi_{m,n_1}(t_1)\psi_{m,n_2}(t_2)$。

通常可定义三个小波，即：

$$\Psi^h(t_1,t_2)=\varphi(t_1)\psi(t_2) \tag{3-329}$$

$$\Psi^v(t_1,t_2)=\psi(t_1)\varphi(t_2) \tag{3-330}$$

$$\Psi^d(t_1,t_2)=\psi(t)\psi(t) \tag{3-331}$$

当 $\Psi^{\lambda}_{m,n}$，$m\in Z,n\in Z^2,\lambda=\{h,v,d\}$ 时，是 $\bigoplus\limits_{m\in Z}W_m=L^2(R^2)$ 的规范正交基，当 m 固定时是 W_m 的规范正交基。

二维离散正交小波以主要解决二维多分辨分析问题。如一个二维函数 $f(t_1,t_2)\in L^2(R^2)$，当分辨率为 m 时，函数 $f(t_1,t_2)$ 的二维小波离散化逼近可以通过内积运算得到，即：

$$\begin{aligned}P^D_m f&=\{\langle f,\Phi_{m,n_1,n_2}\rangle\quad (n_1,n_2)\in Z^2\}\\&=\{\langle f,\varphi_{m,n_1}\varphi_{m,n_2}\rangle\quad (n_1,n_2)\in Z^2\}\end{aligned} \tag{3-332}$$

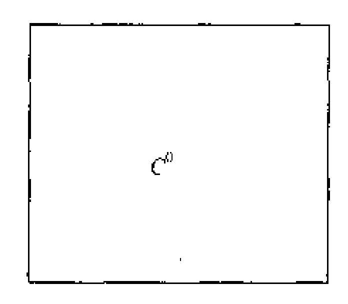
函数的离散化细化逼近可通过 $f(t_1,t_2)$ 与 V_m 补空间 W_m 的规范化正交基向量的内积得到，即

$$\begin{aligned}Q^{D_1}_m f&=\{\langle f,\Psi^h_{m,n_1n_2}\rangle,(n_1,n_2)\in Z^2\}\\&=\{\langle f,\Psi^1_{m,n_1n_2}\rangle,(n_1,n_2)\in Z^2\}\end{aligned} \tag{3-333}$$

$$\begin{aligned}Q^{D_2}_m f&=\{\langle f,\Psi^v_{m,n_1n_2}\rangle,(n_1,n_2)\in Z^2\}\\&=\{\langle f,\Psi^2_{m,n_1n_2}\rangle,(n_1,n_2)\in Z^2\}\end{aligned} \tag{3-334}$$

$$\begin{aligned}Q^{D_3}_m f&=\{\langle f,\Psi^d_{m,n_1n_2}\rangle,(n_1,n_2)\in Z^2\}\\&=\{\langle f,\Psi^3_{m,n_1n_2}\rangle,(n_1,n_2)\in Z^2\}\end{aligned} \tag{3-335}$$

这种逼近过程如图 3-31 所示。



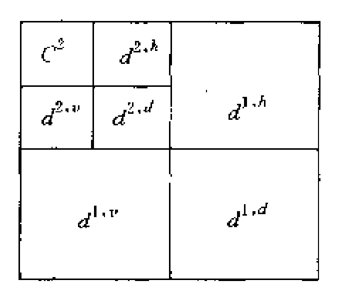


图 3-31 二维离散小波逼近函数 $f(t_1,t_2)$ 原理图

这里把 C^0 看成是原始图像数据，由 $N\times N$ 矩阵组成，第一层为 $\frac{N}{2}\times\frac{N}{2}$，第二层为 $\frac{N}{4}\times\frac{N}{4}$ 等。在广义的情形下，可以把 C^0 看成是一幅图像采样后的二维离散数据，在小波级数展开中，

$f(t_1,t_2)$到 V_m 上的投影 P_mf 是 $f(t_1,t_2)$的一个逼近，它给出了图像的轮廓，$f(t_1,t_2)$到 W_m 上的投影 Q_mf 是 P_mf 到 $P_{m-1}f$ 的细节补充。因此，二维小波基中，

$$\{\Psi_{m,n}^h\}_{n_1n_2\in Z^2} = \{\varphi_{m,n_1}(t_1)\psi_{m,n_2}(t_2)\}_{(n_1,n_2)\in Z^2} \tag{3-336}$$

反映图像水平方向的信息；

$$\{\Psi_{m,n}^v\}_{n_1n_2\in Z^2} = \{\psi_{m,n_1}(t_1)\varphi_{m,n_2}(t_2)\}_{(n_1,n_2)\in Z^2} \tag{3-337}$$

反映图像垂直方向的信息；

$$\{\Psi_{m,n}^d\}_{n_1n_2\in Z^2} = \{\psi_{m,n_1}(t_1)\psi_{m,n_2}(t_2)\}_{(n_1,n_2)\in Z^2} \tag{3-338}$$

反映图像对角线方向的信息。

3.6.7 Mallat 算法

1988 年 Mallat 受到塔式算法的启发在多分辨分析的指导下建立了 Mallat 算法，它的作用可与 FFT 在 Fourier 变换中的作用相提并论。具体算法可描述于下面。

设 V_0 是给定的多分辨率分析尺度空间，因子 $m=0$,，相应的尺度函数为

$$\begin{aligned}\varphi_{m,n}(t) &= 2^{-\frac{m}{2}}\varphi(2^{-m}t-n)\\ &= 2^{-\frac{m}{2}}\sqrt{2}\sum_k h_k\varphi(2^{-m+1}t-2n-k)\\ &= \sum_k h_k\varphi_{m-1,2n+k}(t) = \sum_k h_{k-2n}\varphi_{m-1,n}(t)\end{aligned} \tag{3-339}$$

小波函数为

$$\begin{aligned}\varphi_{m,n}(t) &= 2^{-\frac{m}{2}}\varphi(2^{-m}t-n)\\ &= 2^{-\frac{m}{2}}\sqrt{2}\sum_k g_k\varphi(2^{-m+1}t-2n-k)\\ &= \sum_k g_k\varphi_{m-1,2n+k}(t) = \sum_k g_{k-2n}\varphi_{m-1,k}(t)\end{aligned} \tag{3-340}$$

由 $V_0=V_1\oplus W_1$，若 $f(t)\in V_0$ 则正交小波分解为

$$P_{m-1}f(t) = P_mf(t) + Q_mf(t) \tag{3-341}$$

当 $m=1$ 时，则：

$$P_0f(t) = P_1f(t) + Q_1f(t) \tag{3-342}$$

令：$f^0=P_0f(t)$，$f^1=P_1f(t)$，$d^1=Q_1f(t)=f^0-f^1$，当尺度因子 $m=1$ 转到 $m=2$ 时，则有 $f^0=f^1+d^1$，类似地则有 $f^{m-1}=f^m+d^m$。

由此可得：

$$\begin{aligned}f(t) &= \sum_n a_n^0\varphi(t-n) = 2^{-\frac{1}{2}}\Big[\sum_n a_n^1\varphi(2^{-1}t-n) + \sum_n b_n^1\varphi(2^{-1}t-n)\Big]\\ &= \sum_n a_n^1\sum_k h_k\varphi(t-2n-k) + \sum_n b_n^1\sum_k g_k\varphi(t-2n-k)\\ &= \sum_n a_n^1\sum_j h_{j-2n}\varphi(t-j) + \sum_n b_n^1\sum_j g_{j-2n}\varphi(t-j)\\ &= \Big(\sum_n a_n^1\sum_k h_{j-2n} + \sum_n b_n^1\sum_j g_{j-2n}\Big)\varphi(t-j)\end{aligned} \tag{3-343}$$

代换角标 $n=k$，$j=n$ 则有：

$$f(t) = \Big(\sum_k\sum_n a_k^1h_{n-2k} + \sum_k\sum_n b_n^1g_{n-2k}\Big)\varphi(t-n)$$

$$=\sum_{n}\Big(\sum_{k}a_k^1h_{n-2k}+\sum_{k}b_k^1g_{n-2k}\Big)\varphi(t-n) \tag{3-344}$$

由此可得：

$$a_n^0=\sum_{k}a_k^1h_{n-2k}+\sum_{k}b_k^1g_{n-2k} \tag{3-345}$$

由此推广至系数表达的一般式：

$$a_n^{m-1}=a_n^m+b_n^m=\sum_{k}a_k^mh_{n-2k}+\sum_{k}b_k^mg_{n-2k} \tag{3-346}$$

该式就是由小波系数 a_n^m 和 b_n^m 重构函数 $f(t)$的正交展开式系数 a_n^{m-1} 的公式。

函数 $f(t)$的小波展开系数$\langle f,\varphi_{m,n}\rangle$与$\langle f,\psi_{m,n}\rangle$用双尺度差分公式表示可按下式计算：

$$\begin{aligned}
\langle f,\varphi_{m,n}\rangle &=\int_{-\infty}^{+\infty}f(t)\,\overline{\varphi_{m,n}(t)}\mathrm{d}t \\
&=\int_{-\infty}^{+\infty}f(t)\cdot 2^{-\frac{m}{2}}\,\overline{\varphi(2^{-m}t-n)}\mathrm{d}t \\
&=\int_{-\infty}^{+\infty}f(t)\cdot 2^{-\frac{m}{2}}\cdot\sqrt{2}\sum_{k}\overline{h_k\varphi(2\cdot 2^{-m}t-2n-k)}\mathrm{d}t \\
&=\int_{-\infty}^{+\infty}f(t)\cdot 2^{-\frac{m-1}{2}}\cdot\sum_{k}\overline{h}_k\,\overline{\varphi[2^{-(m-1)}t-(2n+k)]}\mathrm{d}t \\
&=\sum_{j}\overline{h}_{j-2n}\int_{-\infty}^{+\infty}f(t)2^{-\frac{(m-1)}{2}}\,\overline{\varphi[2^{-(m-1)}t-j]}\mathrm{d}t \\
&=\sum_{j}\overline{h}_{j-2n}\int_{-\infty}^{+\infty}f(t)\,\overline{\varphi}_{m-1,j}(t)\mathrm{d}t \\
&=\sum_{j}\overline{h}_{j-2n}\langle f,\varphi_{m-1,j}\rangle \\
&=\sum_{k}\overline{h}_{k-2n}\langle f,\varphi_{m-1,k}\rangle(\text{用 }k\text{ 代 }j)
\end{aligned} \tag{3-347}$$

同理，

$$\langle f,\psi_{m,n}\rangle=\sum_{k}\overline{g}_{k-2n}\langle f,\varphi_{m-1,k}\rangle \tag{3-348}$$

计算步骤可按下列方法进行：

由$\langle f,\varphi_{0,n}\rangle$计算$\langle f,\varphi_{1,n}\rangle$和$\langle f_1,\psi_{1,n}\rangle$，再由$\langle f,\varphi_{1,n}\rangle$计算$\langle f,\varphi_{2,n}\rangle$和$\langle f_1,\psi_{2,n}\rangle$，直至尺度到 m 为止。

由上述计算可知，Mallat 算法本质上不需要知道尺度函数 $\varphi(t)$和小波函数 $\psi(t)$的具体结构，只由系数就可以实现 $f(t)$的分解与重构，因此，称为快速小波变换。

上述的分析从信息处理的观点来看，可认为是一种滤波运算。我们不仿对比一下：

一个线性系数 $h(t)$对信号 $x(t)$的滤波处理可做如下数学描述：

$$y(t)=h(t)*f(t)=\int_{-\infty}^{+\infty}h(t)f(t-\tau)\mathrm{d}\tau \tag{3-349}$$

如果在频域计算则显然有：

$$Y(\omega)=H(\omega)\cdot F(\omega) \tag{3-350}$$

离散化处理，即 $y_n=h_n*f_n=\sum_{k}h_{n-k}f_k$

比较 $a_n^1=\sum_{k}\overline{h}_{k-2n}a_k^0$ 及 $b_n^1=\sum_{k}\overline{g}_{k-2n}a_k^0$，可把它们看成是 a_k^0 与 $\overline{h}_{n-2k}$、$\overline{g}_{n-2k}$的卷积运算，其差别只是对 $\sum_{k}\overline{h}_{n-k}f_k$ 而言是所有的 k 值做卷积，即所谓的“全滤波”。而 $\sum_{k}\overline{h}_{n-2k}a_k^0$ 、$\sum_{k}\overline{g}_{k-2n}b_k^0$

则是 $2n$ 对所有可能的 k 值做卷积，缺少了 n 的奇数$(2n+1)$部分，即为所谓的“半滤波”。比较 $\sum_k h_{n-k}f_k$ 与 $\sum_k a_k^1 h_{n-2k}$，$\sum_k b_k^1 g_{n-2k}$，它们都是离散卷积，也就是滤波处理。其区别只是 n 对 k 的偶数序列$(2k)$作卷积运算，造成 a_k^1、b_k^1 的取值个数比 h_{n-2k}、g_{n-2k} 多一倍，因而称之为“倍滤波”。

令

$$H=\Big(\sum_k h_{n-2k}\Big)_n \qquad G=\Big(\sum_k g_{n-2k}\Big)_n \tag{3-351}$$

$$H^*=\Big(\sum_k \bar{h}_{k-2n}\Big)_n \qquad G^*=\Big(\sum_k \bar{g}_{k-2n}\Big)_n \tag{3-352}$$

则可把滤波处理表示成简洁地形式，即

$$\begin{aligned} a^1 &= H^* a^0, \quad b_1 = G^* a^0 \\ a^0 &= Ha^1 + Gb^1 \\ &= HH^* a^0 + GG^* a^0 \\ &= (HH^* + GG^*)a^0 \end{aligned} \tag{3-353}$$

由此，完全重构 a^0 的条件是 $HH^*+GG^*=I$，H 和 G 是一对共轭正交滤波器组(Conjugate Quadrature Filters)。

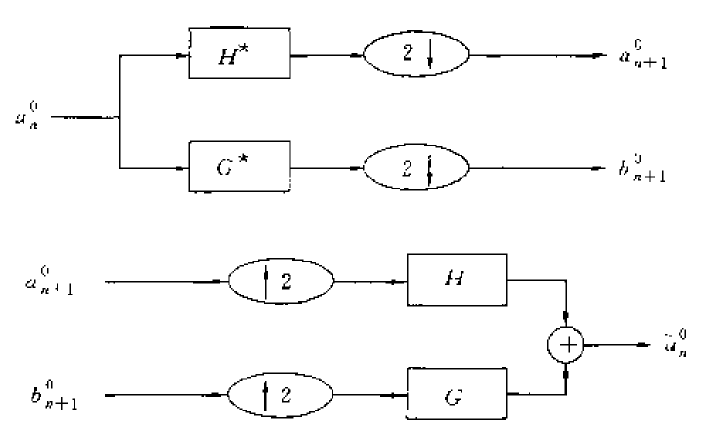
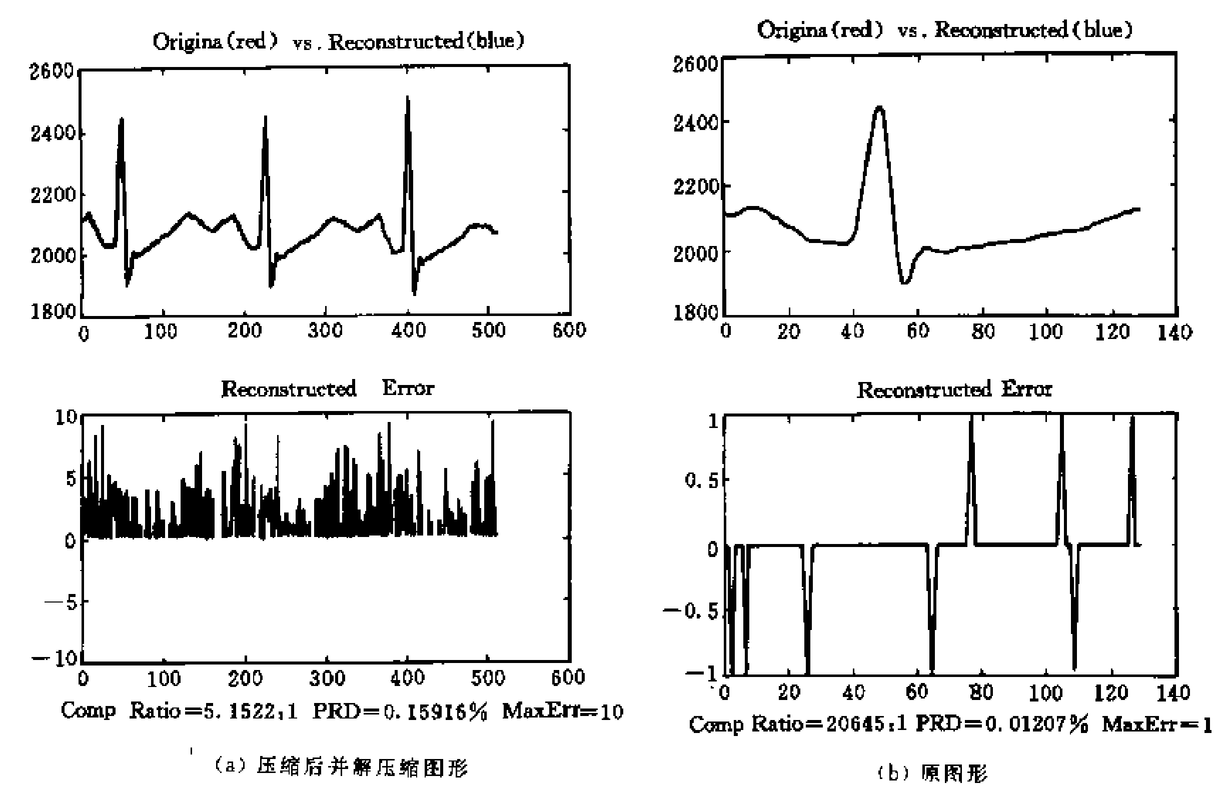


图 3-32　数据分解与重构流图

Mallat 算法可用下图 3-32 来描述，图 3-33 是利用小波对心电图压缩及解压缩的图形。图 3-34 是利用小波进行图像压缩的例子。

上边我们仅对小波理论做了简要的分析，近年来小波理论及应用的研究热情始终有增无减，这促使了小波理论的发展，使其应用越来越广，但是正像 Daubechies 所说的那样，小波本身是一种工具，“它的应用重要的是了解你所研究的问题”。小波理论的奠基人 Meyer 也说过：“小波很时髦，因而引起人们的好奇和兴奋。令人惊奇的是，做为传统的 Fourier 分析的替代物，小波几乎在 20 世纪 80 年代开始同时出现在如下多种多样的领域中：语言的分析与合成，无线电信号的编码，视网膜(Retinian System)系统所进行的信息抽取过程，完全发展的湍流分析，量子理论中的重正化函数，空间内插理论等，可是，如同那些号称使人们能够了解一切的“大综合”一样，这种多学科的自诩性只能使人们不愉快罢了。小波是否将很快与突变理论(Catastroph Theory)或分形理论(Fractal Theory)一起加入那些无所不包系统的百货商店中，在我看来，“小波”的情况稍微有点不一样，因为它们并不构成一种理论而只是一种崭新的科学工具，其实，它们一点也没有用来解释新事物，当 Farge 使用它们来分析湍流模拟结果时，它们起的



（a）压缩后并解压缩图形 （b）原图形

图 3-33 心电图小波压缩的例子

作用差不多与我阅读《Apologide Raimond Sebond》时所戴的眼镜一样。在目前我的年龄所需要的这副眼镜，如果我并不了解 Montaige 的那些思想，我并不会因而放弃不使用它，或者如果我很喜欢那些思想，我也不会因而赞美这副眼镜。对于小波也是一样，它们的恰如其份又是必不可少的作用是帮助我们在各个不同尺度上更好地研究那些复杂现象。”这些认识对我们深入研究并广泛采用小波这一数学工具时，全面认识它的作用不无启迪。



（a）原像 （b）压缩解压缩图像

图 3-34 图像小波变换压缩实例

附录 1　快速傅里叶变换与反变换程序实例

```
#include <math.h>
#include <malloc.h>

#define pi (double)3.14159265359

/*复数定义*/
typedef struct
{
	double re;
	double im;
}COMPLEX;

/*复数加运算*/
COMPLEX Add(COMPLEX c1, COMPLEX c2)
{
	COMPLEX c;
	c.re=c1.re+c2.re;
	c.im=c1.im+c2.im;
	return c;
}

/*复数减运算*/
COMPLEX Sub(COMPLEX c1, COMPLEX c2)
{
	COMPLEX c;
	c.re=c1.re-c2.re;
	c.im=c1.im-c2.im;
	return c;
}

/*复数乘运算*/
COMPLEX Mul(COMPLEX c1, COMPLEX c2)
{
	COMPLEX c;
	c.re=c1.re*c2.re-c1.im*c2.im;
	c.im=c1.re*c2.im+c2.re*c1.im;
	return c;
}

/*快速傅里叶变换
TD为时域值,FD为频域值,power为2的幂数*/
```

```
void FFT(COMPLEX * TD, COMPLEX * FD, int power)
{
	int count;
	int i,j,k,bfsize,p;
	double angle;
	COMPLEX *W, *X1, *X2, *X;

	/* 计算傅里叶变换点数 */
	count=1<<power;
	/* 分配运算所需存储器 */
	W=(COMPLEX *)malloc(sizeof(COMPLEX) * count/2);
	X1=(COMPLEX *)malloc(sizeof(COMPLEX) * count);
	X2=(COMPLEX *)malloc(sizeof(COMPLEX) * count);
	/* 计算加权系数 */
	for(i=0;i<count/2;i++)
	{
		angle=-i * pi * 2/count;
		W[i].re=cos(angle);
		W[i].im=sin(angle);
	}
	/* 将时域点写入存储器 */
	memcpy(X1,TD,sizeof(COMPLEX) * count);
	/* 蝶形运算 */
	for(k=0;k<power;k++)
	{
		for(j=0;j<1<<k;j++)
		{
			bfsize=1<<power-k;
			for(i=0;i<bfsize/2;i++)
			{
				p=j * bfsize;
				X2[i+p]=Add(X1[i+p],X1[i+p+bfsize/2]);
				X2[i+p+bfsize/2]=Mul(Sub(X1[i+p],X1[i+p+bfsize/
				2]),W[i * (1<<k)]);
			}
		}
		X=X1;
		X1=X2;
		X2=X;
	}
	/* 重新排序 */
	for(j=0;j<count;j++)
	{
		p=0;
```

```
            for(i=0;i<power;i++)
            {
                if (j&(1<<(i))) p+=1<<(power-i-1);
            }
            FD[j]=X1[p];
        }
        /* 释放存储器 */
        free(W);
        free(X1);
        free(X2);
}

/* 快速傅里叶反变换,利用快速傅里叶变换
FD 为频域值,TD 为时域值,power 为 2 的幂数 */
void IFFT(COMPLEX *FD, COMPLEX *TD, int power)
{
        int i,count;
        COMPLEX *x;

        /* 计算傅里叶反变换点数 */
        count=1<<power;
        /* 分配运算所需存储器 */
        x=(COMPLEX *)malloc(sizeof(COMPLEX)*count);
        /* 将频域点写入存储器 */
        memcpy(x,FD,sizeof(COMPLEX)*count);
        /* 求频域点的共轭 */
        for(i=0;i<count;i++)
        {
            x[i].im=-x[i].im;
        }
        /* 调用快速傅里叶变换 */
        FFT(x,TD,power);
        /* 求时域点的共轭 */
        for(i=0;i<count;i++)
        {
            TD[i].re/=count;
            TD[i].im=-TD[i].im/count;
        }
        /* 释放存储器 */
        free(x);
}
```

附录 2　快速余弦变换与反变换程序实例

（利用 $2N$ 点傅里叶变换实现快速余弦变换）

```
/* 快速离散余弦变换,利用快速傅里叶变换
```

```
f 为时域值,F 为变换域值,power 为 2 的幂数 */
void DCT(double *f, double *F, int power)
{
        int i,count;
        COMPLEX *X;
        double s;

        /* 计算离散余弦变换点数 */
        count=1<<power;
        /* 分配运算所需存储器 */
        X=(COMPLEX *)malloc(sizeof(COMPLEX)*count*2);
        /* 延拓 */
        memset(X,0,sizeof(COMPLEX)*count*2);
        /* 将时域点写入存储器 */
        for(i=0;i<count;i++)
        {
                X[i].re=f[i];
        }
        /* 调用快速傅里叶变换 */
        FFT(X,X,power+1);
        /* 调整系数 */
        s=1/sqrt(count);
        F[0]=X[0].re*s;
        s*=sqrt(2);
        for(i=1;i<count;i++)
        {
                F[i]=(X[i].re*cos(i*pi/(count*2))-X[i].im*sin(i*pi/
                (count*2)))*s;
        }
        /* 释放存储器 */
        free(X);
}

/* 快速离散余弦反变换,利用快速傅里叶反变换
F 为变换域值,f 为时域值,power 为 2 的幂数 */
void IDCT(double *F, double *f, int power)
{
        int i,count;
        COMPLEX *X;
        double s;

        /* 计算离散余弦反变换点数 */
        count=1<<power;
        /* 分配运算所需存储器 */
```

```
	X=(COMPLEX *)malloc(sizeof(COMPLEX)*count*2);
	/*延拓*/
	memset(X,0,sizeof(COMPLEX)*count*2);
	/*调整频域点并写入存储器*/
	for(i=0;i<count;i++)
	{
		X[i].re=F[i]*cos(i*pi/(count*2));
		X[i].im=F[i]*sin(i*pi/(count*2));
	}
	/*调用快速傅里叶反变换*/
	IFFT(X,X,power+1);
	/*调整系数*/
	s=1/sqrt(count);
	for(i=0;i<count;i++)
	{
		f[i]=(1-sqrt(2))*s*F[0]+sqrt(2)*s*X[i].re*count*2;
	}
	/*释放存储器*/
	free(X);
}
```

附录 3　快速 Walsh-Hadamard 变换与反变换程序实例

```
/*快速沃尔什-哈达玛变换
f 为时域值,F 为变换域值,power 为 2 的幂数*/
void WALh(double *f, double *W, int power)
{
	int count;
	int i,j,k,bfsize,p;
	double *X1, *X2, *X;

	/*计算快速沃尔什变换点数*/
	count=1<<power;
	/*分配运算所需存储器*/
	X1=(double *)malloc(sizeof(double)*count);
	X2=(double *)malloc(sizeof(double)*count);
	/*将时域点写入存储器*/
	memcpy(X1,f,sizeof(double)*count);
	/*蝶形运算*/
	for(k=0;k<power;k++)
	{
		for(j=0;j<1<<k;j++)
		{
```

```
            bfsize=1<<power-k;
            for(i=0;i<bfsize/2;i++)
            {
                p=j*bfsize;
                X2[i+p]=X1[i+p]+X1[i+p+bfsize/2];
                X2[i+p+bfsize/2]=X1[i+p]-X1[i+p+bfsize/2];
            }
        }
        X=X1;
        X1=X2;
        X2=X;
    }
    /*调整系数*/
    for(i=0;i<count;i++)
    {
        W[i]=X1[i]/count;
    }
    /*释放存储器*/
    free(X1);
    free(X2);
}

/*快速沃尔什-哈达玛反变换,利用快速沃尔什-哈达玛变换
F为变换域值,f为时域值,power为2的幂数*/
void IWALh(double *W, double *f, int power)
{
    int i,count;
    count=1<<power;

    /*调用快速沃尔什-哈达玛变换*/
    WALh(W,f,power);
    /*调整系数*/
    for(i=0;i<count;i++)
    {
        f[i]*=count;
    }
}
```

附录4 二维小波分解与重构程序实例

```
//*****(基于Windows的二维小波分解与重构 BC++4.5)*****//
#include <owl/owlpch.h>
#include <owl/applicat.h>
```

```
#include <owl/dc.h>
#include <owl/menu.h>
#include <owl/framewin.h>
#include <owl/scroller.h>
#include <owl/opensave.h>
#include <owl/clipview.h>
#include <string.h>
#include <alloc.h>
#include <dir.h>
#include <math.h>
#include "stdio.h"
#include "stdlib.h"
#include "bmpview.h"

#define PI 3.1415926
#define SIZE 256

#define DD 13
float h[DD]= {-0.00332761,0.00569794,0.0196637,-0.0482603,-0.0485391,
              0.292562,0.564406,0.292562,-0.0485391,-0.0482602,-0.0196637,
              0.00569794,-0.0033276};
float g[DD]= {0.00332761,0.00569794,-0.0196637,-0.0482603,0.0485391,
              0.292562,-0.564406,0.292562,0.0485391,-0.0482602,0.0196637,
              0.00569794,0.0033276};
float hi[DD],gi[DD];

int wavelet_direction=1;

int a(int x,int xsize);
//Threshold//
int s(float x);
//Set Inverse Filter Coefficients//
void coef();
//*****Wavelet Transform*******************//

void wt(int xs,int ys,long xsize,long ysize);
//****Inverse Wavelet Transform***************//
void iwt(int xs,int ys,long xsize,long ysize);

float *img[SIZE],*imgx[SIZE],*imgy[SIZE];

//****for evolution agents*****//
int AgentNum,ActiveAgent;
int SolutionX[1000],SolutionY[1000];
```

```
int SearchNum,AgentLife;
int Agent[1000][2];
int isAgentAlive[1000];
int SearchTime;
unsigned short Image[32][32];


#define MAXAPPNAME 20
static const char AppName[] = "Image Processing";

//
// TBmpViewWindow, a Bitmap displaying window derived from TClipboardViewer to
// facilitate receiving of clipboard change notifications. Could mix it in if
// an additional base was desired.
//
class TSubWindow : public TFrameWindow {
  public:

        TSubWindow(TWindow * parent);
       ~TSubWindow();

  protected:
        void EvSize(UINT sizeType, TSize& size)
              {Invalidate(); TFrameWindow::EvSize(sizeType, size);}
        void Paint(TDC& dc, bool, TRect&);

  DECLARE_RESPONSE_TABLE(TSubWindow);
};

DEFINE_RESPONSE_TABLE1(TSubWindow, TFrameWindow)
  EV_WM_SIZE,
END_RESPONSE_TABLE;

// pointers to different child windows.
//
TWindow * SubWinPtr = 0;
TSubWindow::TSubWindow(TWindow * parent)
  : TFrameWindow(parent)
{
  Attr.Style |= WS_VISIBLE | WS_POPUP | WS_OVERLAPPEDWINDOW,
  Attr.X = 100;
  Attr.Y = 100;
  Attr.W = 300;
  Attr.H = 400;
```

```
//
// Destroy window. SubWinPtr[Type] is set to 0 to indicate that the window
// has be closed.
//
TSubWindow::~TSubWindow()
{
  SubWinPtr = 0;
}
void
TSubWindow::Paint(TDC& dc, bool, TRect&)
{
}


class TBmpViewWindow : virtual public TWindow, public TClipboardViewer {
  public:
        char            FileName[MAXPATH];
        TDib *          Dib;
        TBitmap *       Bitmap;
        TMemoryDC  *    pMemDC;
        TPalette *      Palette;
        TBrush *        BkgndBrush;
        long            Rop;
        int             PixelWidth;
        int             PixelHeight;
        WORD            Colors;
        bool            Fit;
        bool            AutoClipView;
// unsigned char far  * f[256], * g[256];

        TBmpViewWindow();
        ~TBmpViewWindow();
  void ShowSubWindow(TWindow *  parent);
  protected:
        void            CmFileOpen();
        void            CmCopy();
        void            CmPaste();
        void            CmFit();
        void            CmAutoClipView();
        void            CeCopy(TCommandEnabler& ce);
        void            CePaste(TCommandEnabler& ce);
        void            CeFit(TCommandEnabler& ce);
```

```
    void            CeAutoClipView(TCommandEnabler& ce);
    //* * * Image Processing * * //
     void           CmWavelet();
   void             CmAgent();
    //* * * * * * * * * * * * * * * * * * * * * * * *//
    void            Paint(TDC&, bool erase, TRect&);
    void            EvSize(UINT sizeType, TSize&);

    void            EvPaletteChanged(HWND hWndPalChg);
    bool            EvQueryNewPalette();
    void            EvSetFocus(HWND); // should use above when working
    void            EvDrawClipboard();
    void            EvDestroy();

    bool            UpdatePalette(bool alwaysRepaint);
    void            AdjustScroller();
    void            SetCaption(const char * );
    void            SetupFromDib(TDib * dib);
    bool            LoadBitmapFile(const char * );
    bool            LoadBitmapResource(WORD ResId);

  DECLARE_RESPONSE_TABLE(TBmpViewWindow);
};

DEFINE_RESPONSE_TABLE2(TBmpViewWindow, TClipboardViewer, TWindow)
  EV_COMMAND(CM_FILEOPEN, CmFileOpen),
  EV_COMMAND(CM_EDITCOPY, CmCopy),
  EV_COMMAND(CM_EDITPASTE, CmPaste),
  EV_COMMAND(CM_FIT, CmFit),
  EV_COMMAND(CM_AUTOCLIPVIEW, CmAutoClipView),
  EV_COMMAND_ENABLE(CM_EDITCOPY, CeCopy),
  EV_COMMAND_ENABLE(CM_EDITPASTE, CePaste),
  EV_COMMAND_ENABLE(CM_FIT, CeFit),
  EV_COMMAND_ENABLE(CM_AUTOCLIPVIEW, CeAutoClipView),

//* * * * * Image Processing * * * * * * * * * * * * * * //
  EV_COMMAND(CM_WAVELET,CmWavelet),
  EV_COMMAND(CM_AGENT,CmAgent),

//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
  EV_WM_SIZE,
  EV_WM_PALETTECHANGED,
  EV_WM_QUERYNEWPALETTE,
  EV_WM_SETFOCUS,
```

```
  EV_WM_DRAWCLIPBOARD,
  EV_WM_DESTROY,
END_RESPONSE_TABLE;
//* * * * agent search * * * * * * //
//* * * * 小波变换菜单 Wavelet * * * * * * * //
//* * * * 原图像存在 lena.bmp * * * //
//* * * * 运行后,分解图像存在 lenawt.bmp * * * * * * //
//* * * * 重建图像存在 lenaiwt.bmp * * * * * * * * * * * //
void TBmpViewWindow::CmAgent()
{
 char ss[5];
 int i,j,k,x,y;
 AgentNum=5;ActiveAgent=5;SearchNum=0;
 AgentLife=2;
 SearchTime=0;
 for(i=0;i<AgentNum;i++){Agent[i][0]=5*i+1;Agent[i][1]=5*i-1;isAgentAlive[i]=1;}

//                MessageBox("ss","ss",MB_OK);
        if(pMemDC)
        {
              for(x=0;x<32;x++)
               for(y=0;y<32;y++)
                Image[x][y]=(unsigned short)pMemDC->GetPixel(x,y);
//              img[i][j]=f[i][j];
         while(1){
                         MessageBox("ss","ss",MB_OK);
                if(ActiveAgent<=0)break;
               for( i=0;i<AgentNum;i++)
               {
         if(isAgentAlive[i]==1)
         {
                x=Agent[i][0];y=Agent[i][1];
                if(Image[x][y]== 0)
                   {
                         SolutionX[SearchNum]=x;SolutionY[SearchNum]=y;
                         Image[x][y]=255;//* * * marking this point
                       pMemDC->SetPixel(x,y,
                              (unsigned char)Image[x][y]);
                       AdjustScroller();
                         SearchNum+=1;
                         //* * judge direction * * //
            if((x+1)<32&&Image[x+1][y]==0){x=x+1;Agent[i][0]=x;}
            else if((x-1))=0 && Image[x-1][y]==0) {x=x-1; Agent[i][0]=x;}
             else if((y-1))=0&&Image[x][y-1]==0){y=y-1;Agent[i][1]=y;}
```

```
                    else if((y+1)<32&&Image[x][y+1]==0) {y=y+1;Agent[i][1]=y;}
                         else {isAgentAlive[i]=0; ActiveAgent-=1;}
                         }
                   else
                    { //* *judge direction * *//
              if((x+1)<32&&Image[x+1][y]==0){x=x+1;Agent[i][0]=x;}
              else if((x-1))=0 && Image[x-1][y]==0) {x=x-1;Agent[i][0]=x;}
              else if((y-1))-0&&Image[x][y-1]==0){y=y-1;Agent[i][1]=y;}
              else if((y+1)<32&&Image[x][y+1]==0) {y=y+1;Agent[i][1]=y;}
                         else {isAgentAlive[i]=0; ActiveAgent-=1;}
                      }
           SearchTime++;
            }//* *end if isAgentAlive==1
               }//* *end for
         }//* *end while
              sprintf(ss,"%d",SearchTime);
                         MessageBox(ss,"Total Search times",MB_OK);
      }

}

void
TBmpViewWindow::CmWavelet()
{
  unsigned i,j,k;int xs,ys;
  unsigned char buf[SIZE+1],buf1[1078];
  FILE *inputfile, *mfile, *outputfile;
  inputfile=fopen("lena.bmp","rb");
  if(! inputfile)
            { MessageBox("Cannot open file lena.raw", GetApplication()->GetName(), MB_OK);
            CloseWindow(0);
            }
  mfile=fopen("lenawt.bmp","wb");
  if(! mfile)
            { MessageBox("Cannot open file lenawt.raw", GetApplication()->GetName(), MB_OK);
            CloseWindow(0);
            }
  outputfile=fopen("lenaiwt.bmp","wb");
  if(! outputfile)
            { MessageBox("Cannot open file lenawt.raw", GetApplication()->GetName(), MB_OK);
            CloseWindow(0);
            }
  for(i=0;i<=SIZE-1;i++)
      {
```

```
        if((imgx[i]=(float far * )farmalloc(SIZE * sizeof(float)))==NULL)
         { MessageBox("Cannot malloc memory for displaying bitmap file", GetApplication()->Get-
Name(), MB_OK);
        CloseWindow(0);
        }
        if((imgy[i]=(float far * )farmalloc(SIZE * sizeof(float)))==NULL)
         { MessageBox("Cannot malloc memory for displaying bitmap file", GetApplication()->Get-
Name(), MB_OK);
        CloseWindow(0);
        }
        if((img[i]=(float far * )farmalloc(SIZE * sizeof(float)))==NULL)
         { MessageBox("Cannot malloc memory for displaying bitmap file", GetApplication()->Get-
Name(), MB_OK);
        CloseWindow(0);
         }

        }

  //read input file
   fread(buf1,1,1078,inputfile);
      for(i=0;i<SIZE;i++)
   {
       fread(buf,1,SIZE,inputfile);
       for(j=0;j<SIZE;j++)
        img[i][j] = (float)(buf[j])-128.0;
       }
      fclose(inputfile);


 //start wt....
        coef();


       xs=SIZE;
       ys=SIZE;
            wt(xs,ys,SIZE,SIZE);
        xs=xs/2;ys=ys/2;
            wt(xs,ys,SIZE,SIZE);
        xs=xs/2;ys=ys/2;
            wt(xs,ys,SIZE,SIZE);

 //* * * * * write the wt result file * * * * * //
            fwrite(buf1,1,1078,mfile);
        for(i=0;i<SIZE;i++)
        {
```

```
                for(j=0;j<SIZE;j++)
                {
                 buf[j]=(unsigned char)(s(img[i][j])+128);
                 img[i][j]=(float)buf[j]-128.0;
                 }
                 fwrite(buf,1,SIZE,mfile);
            }
            fclose(mfile);

//****start inverse wt....***//
        iwt(xs,ys,SIZE,SIZE);
        xs=2*xs;ys=2*ys;
        iwt(xs,ys,SIZE,SIZE);
        xs=2*xs;ys=2*ys;
        iwt(xs,ys,SIZE,SIZE);
                fwrite(buf1,1,1078,outputfile);
          for(i=0;i<SIZE;i++)
          {
                for(j=0;j<SIZE;j++)
                {
                 buf[j]=(unsigned char)(s(img[i][j])+128);
                 }
          fwrite(buf,1,SIZE,outputfile);
          }
          fclose(outputfile);
/*      LoadBitmapFile("lenawt.bmp");
         SetCaption(strlwr("lenawt.bmp"));
         LoadBitmapFile("lenaiwt.bmp");
         SetCaption(strlwr("lenaiwt.bmp"));
   */
  for(i=0;i<SIZE-1;i++)
  {
        farfree(imgy[i]);
        farfree(imgx[i]);
        farfree(img[i]);
       }
 if(wavelet_direction==1)wavelet_direction=0;
 else wavelet_direction=1;
}
//Edge Process//
int a(int x,int xsize)
{
 if(x<0)x=-x;
 if(x>=xsize)x=xsize*2-x-2;
```

```
 return(x);
 }

//Threshold//
int s(float x)
{
 if(x>127)return(127);
 if(x<-128)return(-128);
 return(x);
 }
//Set Inverse Filter Coefficients//
void coef()
{
 int i;
 for(i=0;i<DD;i++)
  {
     hi[i]=h[DD-1-i];
     gi[i]=g[DD-1-i];
     }
 }
//*********二维小波分解****************//

void wt(int xs,int ys,long xsize,long ysize)
{
 int i,j,k,n;
 float temp1,temp2;
 float bufferx[SIZE],buffery[SIZE];
  for(n=0;n<ys;n++)
 {

  for(i=0;i<xs;i+=2)
     {
      temp1=0;
      temp2=0;
      for(j=-(DD-1)/2;j<=(DD-1)/2;j++)
      temp1=temp1+h[j+(DD-1)/2]*img[n][a(i+j,xs)];
      for(j=-(DD-1)/2+1;j<=(DD-1)/2+1;j++)
      temp2=temp2+g[j+(DD-1)/2-1]*img[n][a(i+j,xs)];
      bufferx[i/2]=temp1;
      bufferx[i/2+xs/2]=temp2;
      }
  for(k=0;k<xs;k++)
     img[n][k]=bufferx[k];
  }
```

```
//*****Filter in the vertical direction*****//
for(n=0;n<xs;n++)
{

 for(i=0;i<ys;i+=2)
      {
       temp1=0;
       temp2=0;
       for(j=-(DD-1)/2;j<=(DD-1)/2;j++)
       temp1=temp1+h[j+(DD-1)/2]*img[a(i+j,ys)][n];
       for(j=-(DD-1)/2+1;j<=(DD-1)/2+1;j++)
       temp2=temp2+g[j+(DD-1)/2-1]*img[a(i+j,ys)][n];
       buffery[i/2]=temp1;
       buffery[i/2+ys/2]=temp2;
       }
 for(k=0;k<ys;k++)
      img[k][n]=buffery[k];
 }
}
//*****二维小波重构*******************//
void iwt(int xs,int ys,long xsize,long ysize)
{
 int i,j,k,n;
 float temp1,temp2;
 float buffer1[SIZE], buffer2[SIZE];
 //**Filter in the vertical direction******//
 for(n=0;n<xs;n++)
 {
  for(k=0;k<ys/2;k++)
  {
      buffer1[k*2]=img[k][n];
      buffer1[k*2+1]=0;
      }
  for(i=0;i<ys;i++)
      {
       temp1=0;
       for(j=-(DD-1)/2;j<=(DD-1)/2;j++)
       temp1=temp1+hi[j+(DD-1)/2]*buffer1[a(i+j,ys)];
       buffer2[i]=temp1;
      }
  for(k=0;k<ys/2;k++)
  {
      buffer1[k*2]=img[k+ys/2][n];
      buffer1[k*2+1]=0;
```

```
        }
    for(i=0;i<ys;i++)
    {
          temp1=0;
          for(j=-(DD-1)/2-1;j<=(DD-1)/2-1;j++)
          temp1=temp1+gi[j+(DD-1)/2+1]*buffer1[a(i+j,ys)];
          temp2=temp1+buffer2[i];
          buffer2[i]=temp2;
        }
    for(k=0;k<ys;k++)
        img[k][n]=buffer2[k];
  }
  //*****Filter in the horizontal direction*****//
  for(n=0;n<ys;n++)
  {
    for(k=0;k<xs/2;k++)
    {
        buffer1[k*2]=img[n][k];
        buffer1[k*2+1]=0;
        }
    for(i=0;i<xs;i++)
    {
        temp1=0;
        for(j=-(DD-1)/2;j<=(DD-1)/2;j++)
        temp1=temp1+hi[j+(DD-1)/2]*buffer1[a(i+j,xs)];
        buffer2[i]=temp1;
        }
    for(k=0;k<xs/2;k++)
    {
        buffer1[k*2]=img[n][k+xs/2];
        buffer1[k*2+1]=0;
        }
    for(i=0;i<xs;i++)
    {
        temp1=0;
        for(j=-(DD-1)/2-1;j<=(DD-1)/2-1;j++)
        temp1=temp1+gi[j+(DD-1)/2+1]*buffer1[a(i+j,xs)];
        temp2=temp1+buffer2[i];
        buffer2[i]=temp2;
        }
    for(k=0;k<xs;k++)
      img[n][k]=buffer2[k]*4;
  }
```

```
//
// Constructor for a TBmpViewWindow, sets scroll styles and constructs
// the Scroller object. Also sets the Rop based on whether the display
// is monochrome (two-color) or polychrome.
//
void
TBmpViewWindow::ShowSubWindow(TWindow * parent)
{
  if (! SubWinPtr)
        (SubWinPtr = new TSubWindow(parent))->Create();
  else {
        SubWinPtr->SetFocus();
        SubWinPtr->SetActiveWindow();

        }
  SubWinPtr->Invalidate(1);
}



TBmpViewWindow::TBmpViewWindow()
  : TWindow(0, 0, 0), TClipboardViewer()
{
  // Init TWindow since TClipboardViewer is usually a mixin and doesn't do it
  //
// Init(0, 0, 0);

  Attr.Style |= WS_VSCROLL | WS_HSCROLL;
  Attr.W= 640;Attr.H=480;

  Dib = 0;
  Bitmap = 0;
  pMemDC = 0;
  Palette = 0;
  BkgndBrush = new TBrush(::GetSysColor(COLOR_WINDOW));
  Scroller = new TScroller(this, 1, 1, 200, 200);
  Fit = FALSE;
  AutoClipView = FALSE;

  TScreenDC screenDC;
  if (screenDC.GetDeviceCaps(NUMCOLORS) < 3 )
       Rop = NOTSRCCOPY;
  else
       Rop = SRCCOPY;
```

```
  SetCaption(0);
}

TBmpViewWindow::~TBmpViewWindow()
{
if(Bitmap! =0)
  delete Bitmap;
if(Palette! =0)
  delete Palette;
if(Dib! =0)
  delete Dib;
if(pMemDC! =0)
  delete pMemDC;
  delete BkgndBrush;
}

//
// Build up a caption based on a filename, and set it into the title.
//
void
TBmpViewWindow::SetCaption(const char * name)
{
  char caption[MAXPATH + MAXAPPNAME + 2 + 1];
  strcpy(FileName, name ? name : "(Untitled)");
  strcpy(caption, GetApplication()->GetName());
  strcat(caption, " - ");
  strcat(caption, FileName);
  if (Parent)
       Parent->SetCaption(caption);
}

//
// Make a metafile & put it on the clipboard.
// Make a copy of each of the objects & place the copies on the clipboard
//
void
TBmpViewWindow::CmCopy()
{
}

void
TBmpViewWindow::CeCopy(TCommandEnabler& ce)
{
  ce.Enable(Dib != 0);
```

```
//
// When a user selects edit.paste, get the data from the clipboard.  This
// routine prefers CF_META over CF_DIB over CF_BITMAP
//
void
TBmpViewWindow::CmPaste()
{
}

void
TBmpViewWindow::CePaste(TCommandEnabler& ce)
{
}

//
//
// Toggle Fit member variable & adjust scroller as needed
//
void
TBmpViewWindow::CmFit()
{
  Fit = ! Fit;
  AdjustScroller();
}

//
// The fit menu item is checked if the Fit member is true
//
void
TBmpViewWindow::CeFit(TCommandEnabler& ce)
{
  ce.SetCheck(Fit ? TCommandEnabler::Checked : TCommandEnabler::Unchecked);
}

//
// Toggle AutoAutoClipView member variable
//
void
TBmpViewWindow::CmAutoClipView()
{
  AutoClipView = ! AutoClipView;
}
```

```
//
// Check AutoClipView according to flag
//
void
TBmpViewWindow::CeAutoClipView(TCommandEnabler& ce)
{
  ce.SetCheck(AutoClipView ? TCommandEnabler::Checked : TCommandEnabler::Unchecked);
}


void
TBmpViewWindow::CmFileOpen()
{
  static TOpenSaveDialog::TData data (
      OFN_HIDEREADONLY|OFN_FILEMUSTEXIST|OFN_NOREADONLYRETURN,
      "Bitmap Files (*.BMP)|*.bmp|",
      0,
      "",
      "BMP"
  );
  if (TFileOpenDialog(this, data).Execute() == IDOK) {
      char fileTitle[MAXPATH];
      TOpenSaveDialog::GetFileTitle(data.FileName, fileTitle, MAXPATH);
      LoadBitmapFile(fileTitle);
      SetCaption(strlwr(fileTitle));
  }
}

//
// Adjust the Scroller range so that the the origin is the
// upper-most scrollable point and the corner is the
// bottom-most.
//
void
TBmpViewWindow::AdjustScroller()
{
  TRect clientRect = GetClientRect();

  // only show scrollbars when img is larger than
  // the client area and we are not stretching to fit.
  //
  if (Fit)
      Scroller->SetRange(0, 0);
```

```
  else {
        TPoint Range(Max(PixelWidth-clientRect.Width(), 0),
                                        Max(PixelHeight-clientRect.Height(), 0));
        Scroller->SetRange(Range.x, Range.y);
  }
  Scroller->ScrollTo(0, 0);
 // if (! GetUpdateRect(clientRect, FALSE))
        Invalidate(TRUE);
}

//
// Reset scroller range.
//
void
TBmpViewWindow::EvSize(UINT SizeType, TSize& Size)
{
  TWindow::EvSize(SizeType, Size);
  if (SizeType != SIZEICONIC) {
        AdjustScroller();
        Invalidate(FALSE);
  }
}

//
// Somebody changed the palette. If its not us, then we need to update.
//
void
TBmpViewWindow::EvPaletteChanged(HWND hWndPalChg)
{
  if (hWndPalChg != HWindow)
        UpdatePalette(TRUE); // pass FALSE to UpdateColors() instead of repaint
}

//
// We need to re-realize the logical palette each time
// we regain the input focus
//
bool
TBmpViewWindow::EvQueryNewPalette()
{
  return UpdatePalette(TRUE);
}
```

```
//
// Use this message temporarily until the palette msgs get routed to us
//
void
TBmpViewWindow::EvSetFocus(HWND)
{
  UpdatePalette(TRUE);
}

void
TBmpViewWindow::EvDrawClipboard()
{
  if (TClipboardViewer::DoDrawClipboard() == esComplete)
      return;
  if (AutoClipView)
      CmPaste();
}

void
TBmpViewWindow::EvDestroy()
{
  if (TClipboardViewer::DoDestroy() == esComplete)
      return;
  TWindow::EvDestroy();
}

bool
TBmpViewWindow::UpdatePalette(bool alwaysRepaint)
{
  if (Palette) {
      TClientDC clientDC(*this);
      #if !defined(__WIN32__)
          Palette->UnrealizeObject();
      #endif
// TRACEX(BmpView, 2, "UpdatePalette::Palette: " << hex << (UINT)(HPALETTE)*Palette);
      clientDC.SelectObject(*Palette, FALSE);
      if (clientDC.RealizePalette() > 0)
          if (alwaysRepaint)
            Invalidate(FALSE);
          else
            clientDC.UpdateColors();
// TRACEX(BmpView, 2, "Leave UpdatePalette(TRUE)");
      return TRUE;
  }
```

```
 // TRACEX(BmpView, 2, "Leave UpdatePalette(FALSE)");
  return FALSE;
}

//
// Responds to an incoming Paint message by redrawing the bitmap.
// The Scroller's BeginView method, which sets the viewport origin
// relative to the present scroll position, has been called by TWindow's
// EvPaint
// Note that we Invalidate() ourselves with false to avoid the background
// paint flicker. Thats why we use
//
// The code can use either the Bitmap member with Stretch- and Bit- Blts,
// or use the Dib member with Stretch- and Set- DIBits...
//
void
TBmpViewWindow::Paint(TDC& dc, bool, TRect&)
{
  TRect clientRect = GetClientRect();
  dc.SelectObject( * BkgndBrush);

  if (Bitmap)
      {
            if(pMemDC)
              if (Palette) {
                        dc.SelectObject( * Palette, FALSE);
                        dc.RealizePalette();
                        pMemDC->SelectObject( * Palette, FALSE);
                              }



             Rop=SRCCOPY;

      TRect imgRect(0,0, PixelWidth, PixelHeight);
//    dc.PatBlt(imgRect, PATCOPY);
//    for(int j=0;j<=128;j++)
//    for(int i=0;i<=255;i++)pMemDC->SetPixel(j,i,TColor(120));
      dc.BitBlt(imgRect, * pMemDC, TPoint(0,0), Rop);

       }

}

void
```

```
TBmpViewWindow::SetupFromDib(TDib * dib)
{
  if(Dib! =0)delete Dib;
  Dib = dib;
  if(Palette! =0)delete Palette;
  Palette = new TPalette( * Dib);
  if(Bitmap! =0)delete Bitmap;
  Bitmap = new TBitmap( * Dib, Palette);

        if(pMemDC! =0)delete pMemDC;
        pMemDC=new TMemoryDC;
       ((TMemoryDC * )pMemDC)->SelectObject( * Bitmap);
  PixelWidth = Dib->Width();
  PixelHeight = Dib->Height();

 //* * * * Change the Image Pixels * * * * * //
 // unsigned long TotalPixel=PixelWidth * PixelHeight-1;
 // unsigned char far * buf;
 // if((buf= (unsigned char far * )farmalloc(TotalPixel))= =NULL)
 // { MessageBox ("Cannot malloc memory for displaying bitmap file", GetApplication()->GetName(),
MB _OK);
 // CloseWindow(0);
 // }
 // for(unsigned long i=0;i<=(unsigned long)65535;i++)buf[i]=0;
// Bitmap->SetBitmapBits((unsigned long)65535,buf);
// farfree(buf);
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
  UpdatePalette(TRUE);
  AdjustScroller();
#if defined(PAINT _ICON)
  Parent->SetIcon(0, 0); // enable to paint the icon from the bitmap
#endif
}

//
// Test if the passed resource is a Windows 3. 0 (or PM 1. x) DI bitmap
// and if so read it.
// Report errors if unable to do so. Adjust the Scroller to the new
// bitmap dimensions.
//
bool
TBmpViewWindow::LoadBitmapResource(WORD resId)
{
  TDib * newDib;
```

```
        newDib = new TDib( * GetModule(), TResId(resId));

   SetupFromDib(newDib);
   return TRUE;
 }

//
// Test if the passed file is a Windows 3.0 DI (or PM 1.x) bitmap
// and if so read it.
// Report errors if unable to do so. Adjust the Scroller to the new
// bitmap dimensions.
//
bool
TBmpViewWindow::LoadBitmapFile(const char * name)
{
   TDib * newDib;
        newDib = new TDib(name);

          SetupFromDib(newDib);
          return TRUE;
}


//--------------------------------------------------------------------------------------------------------

class TBmpViewApp : public TApplication {
  public:
        TBmpViewApp(const char far * name) : TApplication(name) {}
        void InitMainWindow() {
              MainWindow = new TFrameWindow(0, Name, new TBmpViewWindow);
              MainWindow->AssignMenu("MainMenu");
              MainWindow->SetIcon(this, "MainIcon");
        }
};

//--------------------------------------------------------------------------------------------------------

int
OwlMain(int /* argc */, char * /* argv */ [])
{
  return TBmpViewApp(AppName).Run();
}
```

思 考 题

1. 函数 $f(x,y)$可作 Fourier 变换的基本条件是什么？

2. 连续的非周期函数的 Fourier 谱是什么样的谱？连续的周期函数的 Fourier 谱又是什么样的谱？

3. 二维 Fourier 变换有哪些性质？

4. 试证明 Fourier 变换的旋转性质。

5. 离散 Fourier 变换有哪些性质？

6. 试证明离散 Fourier 变换的卷积定理 $x(n) \cdot y(n) \Leftrightarrow X(m) * Y(m)$。

7. FFT 的基本思想是什么？

8. 当 $N=16$ 时，试画出 FFT 的两种流程图？

9. 试写出 128×128 大小的图像的 FFT 程序？并上机实验。

10. 基 2 的时间分解和基 2 的频率分解的要点是什么？

11. 用计算机实现 FFT 要解决哪些问题？如何解决？

12. 如果 $N=16$，试求 $x_3(2)$的对偶节点。

13. 如果 $N=16$，试求 $x_2(9)$的对偶节点。

14. 试求 $N=16$，$x_3(4)$的加权系数 W_{16}^{p}。

15. 试求 $N=16$，$x_2(8)$的加权系数 W_{16}^{p}。

16. 如何实现离散余弦变换的快速算法？

17. 试写出利用 FFT 实现的离散余弦变换的程序，并上机实验。

18. 沃尔什函数是如何定义的？它们之间的关系是什么？

19. 如何用 Rademacher 函数来构造三种沃尔什函数？

20. 如果 $p=4$，试求 $\mathrm{wal_W}(9,t)$的 Rademacher 函数表达式。

21. 如果 $p=4$，试求 $\mathrm{wal_P}(9,t)$的 Rademacher 函数表达式。

22. 如果 $p=4$，试求 $\mathrm{wal_H}(9,t)$的 Rademacher 函数表达式。

23. 如果 $\mathrm{wal_W}(5,t)$已知，试求与其相对应的 $\mathrm{wal_P}(i,t)$及 $\mathrm{wal_H}(i,t)$？

24. 离散沃尔什变换有哪些性质？

25. 何为模 2 移位序列？它是如何构造的？

26. 试证明模 2 移位列率卷积定理。

27. 何为循环移位序列？它是如何构造的？

28. 快速沃尔什变换的基本思想是什么？

29. 当 $N=16$ 时，试画出快速沃尔什变换的两种流程图？

30. 何为全域函数及区域函数？

31. 如何由$\frac{N}{2}$阶斜矩阵构造 N 阶斜矩阵？

32. Gabor 变换是如何定义的？它是如何产生的？

33. 什么是 Heisenberg 测不准原理？它总结了传统信号分析的什么限制？

34. Gabor 变换的特点是什么？

35. 小波变换是如何定义的？小波函数是唯一的吗？

36. 一个小波函数应满足哪些容许性条件？

37. 何为 Haar 小波？何为 Morlet 小波？何为 Mexico Hat 小波？
38. 试述小波变换的基本性质。
39. 离散小波是如何定义的？
40. 什么是尺度函数？它对小波构造有何意义？
41. 试述紧支集的概念。
42. 试述框架的概念。
43. Daubechies 小波的精髓是什么？
44. 如何构造 B 样条小波？
45. 什么是小波包？它对小波分析有何意义？
46. 建立小波包库的常用准则是什么？
47. 试述二维连续小波的定义。
48. 二维连续小波的容许性条件是什么？
49. 什么是 Mallat 算法？它的意义何在？
50. 如何实现 Mallat 算法？

第4章　图像增强

图像增强是数字图像处理的基本内容之一。

图像增强是指按特定的需要突出一幅图像中的某些信息，同时，削弱或去除某些不需要的信息的处理方法。其主要目的是使处理后的图像对某种特定的应用来说，比原始图像更适用。因此，这类处理是为了某种应用目的而去改善图像质量的。处理的结果使图像更适合于人的视觉特性或机器的识别系统。应该明确的是增强处理并不能增强原始图像的信息，其结果只能增强对某种信息的辨别能力，而这种处理有可能损失一些其他信息。

图像增强技术主要包括直方图修改处理，图像平滑化处理，图像尖锐化处理及彩色处理技术等。在实用中可以采用单一方法处理，也可以采用几种方法联合处理，以便达到预期的增强效果。

图像增强技术基本上可分成两大类：一类是频域处理法，一类是空域处理法。

频域处理法的基础是卷积定理。它采用修改图像傅里叶变换的方法实现对图像的增强处理。由卷积定理可知，如果原始图像是 $f(x,y)$，处理后的图像是 $g(x,y)$，而 $h(x,y)$ 是处理系统的冲激响应，那么，处理过程可由下式表示：

$$g(x,y) = h(x,y) * f(x,y) \tag{4-1}$$

其中 * 代表卷积。如果 $G(u,v)$，$H(u,v)$，$F(u,v)$，分别是 $g(x,y)$，$h(x,y)$，$f(x,y)$ 的傅里叶变换。那么，上面的卷积关系可表示为变换域的乘积关系，即

$$G(u,v) = H(u,v) \cdot F(u,v) \tag{4-2}$$

式中，$H(u,v)$ 为传递函数。

在增强问题中，$f(x,y)$ 是给定的原始数据，经傅里叶变换可得到 $F(u,v)$。选择合适的 $H(u,v)$，使得由式

$$g(x,y) = \mathscr{F}^{-1}[H(u,v) \cdot F(u,v)] \tag{4-3}$$

得到的 $g(x,y)$ 比 $f(x,y)$ 在某些特性方面更加鲜明、突出，因而更加易于识别、解译。例如，可以强调图像中的低频分量使图像得到平滑，也可以强调图像中的高频分量使图像的边缘得到增强等等。以上就是频域处理法的基本原理。

空域法是直接对图像中的像素进行处理，基本上是以灰度映射变换为基础的。所用的映射变换取决于增强的目的。例如增加图像的对比度，改善图像的灰度层次等处理均属空域法处理。

应该特别提及的是增强后的图像质量好坏主要靠人的视觉来评定，而视觉评定是一种高度的主观处理。因此，为了一种特定的用途而采用的一种特定的处理方法，得到一幅特定的图像，对其质量的评价方法和准则也是特定的，所以，很难对各种处理定出一个通用的标准。由此可知，图像增强没有通用理论。

4.1 用直方图修改技术进行图像增强

灰度级的直方图描述了一幅图像的概貌，用修改直方图的方法增强图像是实用而有效的处理方法之一。

4.1.1 直方图

什么是灰度级的直方图呢？简单地说，灰度级的直方图就是反映一幅图像中的灰度级与出现这种灰度的概率之间的关系的图形。

设变量 r 代表图像中像素灰度级。在图像中，像素的灰度级可作归一化处理，这样，r 的值将限定在下述范围之内：

$$0 \leqslant r \leqslant 1 \tag{4-4}$$

在灰度级中，$r=0$ 代表黑，$r=1$ 代表白。对于一幅给定的图像来说，每一个像素取得[0,1]区间内的灰度级是随机的，也就是说 r 是一个随机变量。假定对每一瞬间它们是连续的随机变量，那么，就可以用概率密度函数 $p_r(r)$ 来表示原始图像的灰度分布。如果用直角坐标系的横轴代表灰度级 r，用纵轴代表灰度级的概率密度函数 $p_r(r)$，这样就可以针对一幅图像在这个坐标系中作一曲线来。这条曲线在概率论中就是分布密度曲线(见图 4-1)。

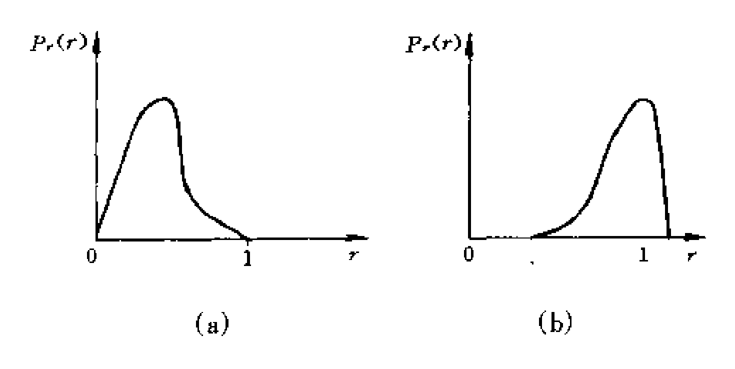


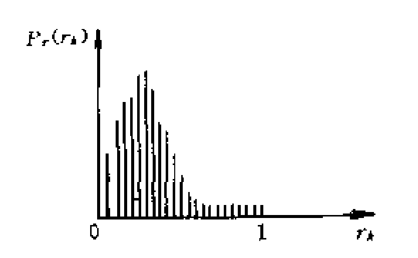
图 4-1 图像灰度分布概率密度函数

从图像灰度级的分布可以看出一幅图像的灰度分布特性。例如，从图 4-1 中的(a)和(b)两个灰度密度分布函数中可以看出：(a)的大多数像素灰度值取在较暗的区域，所以这幅图像肯定较暗，一般在摄影过程中曝光过强就会造成这种结果；而(b)图像的像素灰度值集中在亮区，因此，图像(b)的特性将偏亮，一般在摄影中曝光太弱将导致这种结果。当然，从两幅图像的灰度分布来看图像的质量均不理想。

为了有利于数字图像处理，必须引入离散形式。在离散形式下，用 r_k 代表离散灰度级，用 $P_r(r_k)$ 代表 $p_r(r)$，并且有下式成立：

$$P_r(r_k) = \frac{n_k}{n} \qquad 0 \leqslant r_k \leqslant 1$$
$$k = 0,1,2,\cdots,l-1 \tag{4-5}$$

式中 n_k 为图像中出现 r_k 这种灰度的像素数，n 是图像中像素总数，而 $\frac{n_k}{n}$ 就是概率论中所说的频数。在直角坐标系中作出 r_k 与 $P_r(r_k)$ 的关系图形，这个图形称为直方图。如图 4-2 所示。



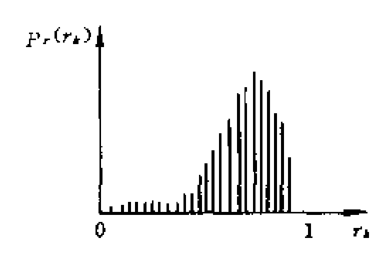


图 4-2 灰度级的直方图

4.1.2 直方图修改技术的基础

如上面所述，一幅给定的图像的灰度级分布在 $0\leqslant r\leqslant 1$ 范围内。可以对[0, 1]区间内的任一个 r 值进行如下变换：

$$s = T(r) \tag{4-6}$$

也就是说，通过上述变换，每个原始图像的像素灰度值 r 都对应产生一个 s 值。变换函数 $T(r)$ 应满足下列条件：

1）在 $0\leqslant r\leqslant 1$ 区间内，$T(r)$单值单调增加；

2）对于 $0\leqslant r\leqslant 1$，有 $0\leqslant T(r)\leqslant 1$。

这里的第一个条件保证了图像的灰度级从白到黑的次序不变。第二个条件则保证了映射变换后的像素灰度值在允许的范围内。满足这两个条件的变换函数的一个例子如图 4-3 所示。

从 s 到 r 的反变换可用式(4-7)表示：

$$r = T^{-1}(s) \tag{4-7}$$

由概率论理论可知，如果已知随机变量 ξ 的概率密度为 $p_r(r)$，而随机变量 η 是 ξ 的函数，即 $\eta=T(\xi)$，η 的概率密度为 $p_s(s)$，所以可以由 $p_r(r)$求出 $p_s(s)$。

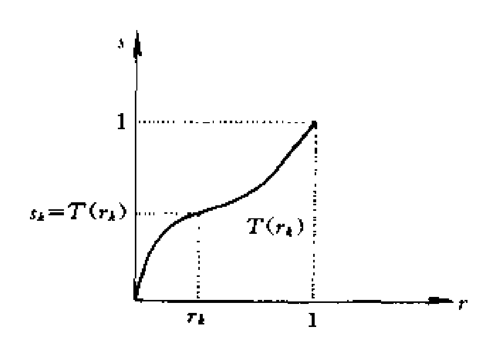




图 4-3 一种灰度变换函数

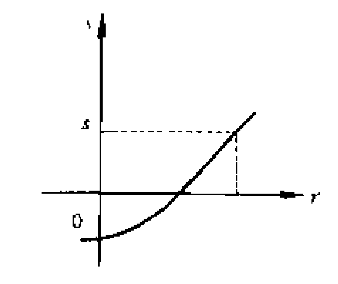


图 4-4 r 和 s 的变换函数关系

因为 $s=T(r)$是单调增加的，由数学分析可知，它的反函数 $r=T^{-1}(s)$也是单调函数。在这种情况下，如图 4-4 所示，$\eta<s$ 且仅当 $\xi<r$ 时发生，所以可以求得随机变量 η 的分布函数为

$$F_\eta(s) = p(\eta < s) = p[\xi < r] = \int_{-\infty}^{r} p_r(x)\mathrm{d}x \tag{4-8}$$

对式(4-8)两边求导，即可得到随机变量 η 的分布密度函数 $p_s(s)$为

$$p_s(s) = p_r(r)\cdot\frac{\mathrm{d}}{\mathrm{d}s}[T^{-1}(s)] = \left[p_r(r)\cdot\frac{\mathrm{d}r}{\mathrm{d}s}\right]_{r=T^{-1}(s)} \tag{4-9}$$

通过变换函数 $T(r)$ 可以控制图像灰度级的概率密度函数，从而改变图像的灰度层次。这就是直方图修改技术的基础。

4.1.3 直方图均衡化处理

直方图均衡化处理是以累积分布函数变换法为基础的直方图修正法。假定变换函数为

$$s = T(r) = \int_0^r p_r(\omega)\mathrm{d}\omega \tag{4-10}$$

式中 ω 是积分变量，而 $\int_0^r p_r(\omega)\mathrm{d}\omega$ 就是 r 的累积分布函数(CDF)。这里，累积分布函数是 r 的函数，并且单调地从 0 增加到 1，所以这个变换函数满足关于 $T(r)$ 在 $0\leqslant r\leqslant 1$ 内单值单调增加，在 $0\leqslant r\leqslant 1$ 内有 $0\leqslant T(r)\leqslant 1$ 的两个条件。

对式(4-10)中的 r 求导，则

$$\frac{\mathrm{d}s}{\mathrm{d}r} = p_r(r) \tag{4-11}$$

再把结果代入式(4-9)则

$$\begin{aligned} p_s(s) &= \left[p_r(r)\cdot\frac{\mathrm{d}r}{\mathrm{d}s}\right]_{r=T^{-1}(s)} \\ &= \left[p_r(r)\cdot\frac{1}{\frac{\mathrm{d}s}{\mathrm{d}r}}\right]_{r=T^{-1}(s)} = \left[p_r(r)\cdot\frac{1}{p_r(r)}\right] = 1 \end{aligned} \tag{4-12}$$

由上面的推导可见，在变换后的变量 s 的定义域内的概率密度是均匀分布的。由此可见，用 r 的累积分布函数作为变换函数可产生一幅灰度级分布具有均匀概率密度的图像。其结果扩展了像素取值的动态范围。

例如，在图 4-5 中，(a)是原始图像的概率密度函数。从图中可知，这幅图像的灰度集中在较暗的区域，这相当于一幅曝光过强的照片。由图(a)可知，原始图像的概率密度函数为

$$p_r(r) = \begin{cases} -2r-2 & 0\leqslant r\leqslant 1 \\ 0 & \text{其他值} \end{cases}$$

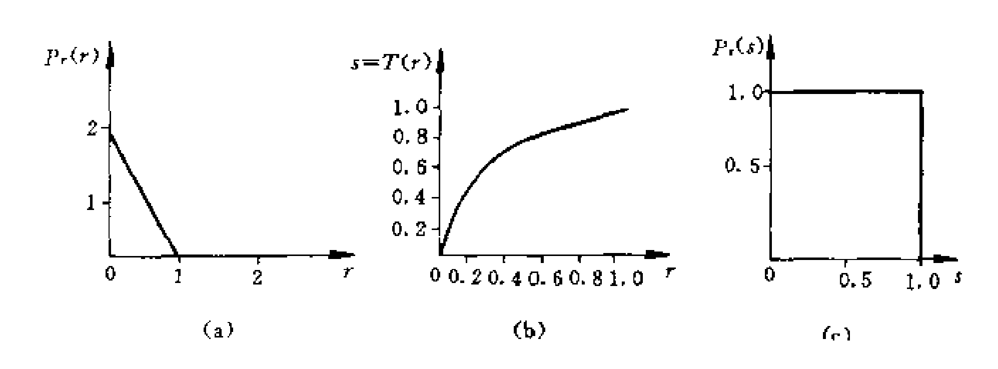


图 4-5 均匀密度变换法

用累积分布函数原理求变换函数：

$$s = T(r) = \int_0^r p_r(\omega)\mathrm{d}\omega$$

$$= \int_0^r (-2\omega + 2)\mathrm{d}\omega = -r^2 + 2r$$

由此可知变换后的 s 值与 r 值的关系为

$$s = -r^2 - 2r = T(r)$$

按照这样的关系变换就可以得到一幅改善了质量的新图像。这幅图像的灰度层次将不再是呈现黑暗色调的图像，而是一幅灰度层次较为适中的，比原始图像清晰，明快得多的图像。

下面还可以通过简单的推证，证明变换后的灰度级概率密度是均匀分布的。

因为 $$s = T(r) = -r^2 + 2r$$

所以 $$r = T^{-1}(r) = 1 \pm \sqrt{1-s}$$

由于 r 取值在[0, 1]区间内，所以

$$r = 1 - \sqrt{1-s}$$

$$\frac{\mathrm{d}r}{\mathrm{d}s} = \frac{\mathrm{d}}{\mathrm{d}s}[1 - \sqrt{1-s}] = \frac{1}{2\sqrt{1-s}}$$

而 $$p_r(r) = -2r + 2 = -2(1 - \sqrt{1-s}) + 2 = 2\sqrt{1-s}$$

因此 $$p_s(s) = \left[p_r(r) \cdot \frac{\mathrm{d}r}{\mathrm{d}s}\right]_{r=T^{-1}(s)} = \left[2\sqrt{1-s} \cdot \frac{1}{2\sqrt{1-s}}\right] = 1$$

这个简单的证明说明在希望的灰度级范围内，它是均匀密度。

图 4-5 (b)和(c)分别为变换函数和变换后的均匀的概率密度函数。

上面的修正方法是以连续随机变量为基础进行讨论的。正如前面谈到的那样，为了对图像进行数字处理，必须引入离散形式的公式。当灰度级是离散值的时候，可用频数近似代替概率值，即

$$P_r(r_k) = \frac{n_k}{n} \qquad 0 \leqslant r_k \leqslant 1 \qquad k = 0, 1, 2, \cdots, l-1 \tag{4-13}$$

式中 l 是灰度级的总数目，$P_r(r_k)$是取第 k 级灰度值的概率，n_k 是在图像中出现第 k 级灰度的次数，n 是图像中像素总数。通常把为得到均匀直方图的图像增强技术叫做直方图均衡化处理或直方图线性化处理。

式(4-10)的离散形式可由式(4-14)表示：

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} P_r(r_j) \qquad 0 \leqslant r_j \leqslant 1 \qquad k = 0, 1, 2, \cdots, l-1 \tag{4-14}$$

其反变换式为

$$r_k = T^{-1}(s_k) \tag{4-15}$$

例如假定有一幅像素数为 64×64，灰度级为 8 级的图像，其灰度级分布如表 4-1 所示，对其进行均衡化处理。其灰度级直方图如图 4-5 所示。

表 4-1　64×64 大小的图像灰度分布表

r_k	n_k	$P_r(r_k)=\frac{n_k}{n}$
$r_0=0$	790	0.19
$r_1=\frac{1}{7}$	1023	0.25
$r_2=\frac{2}{7}$	850	0.21
$r_3=\frac{3}{7}$	656	0.16
$r_4=\frac{4}{7}$	329	0.08
$r_5=\frac{5}{7}$	245	0.06
$r_6=\frac{6}{7}$	122	0.03
$r_7=1$	81	0.02

处理过程如下：

由式(4-14)可得到变换函数：

$$s_0=T(r_0)=\sum_{j=0}^{0}P_r(r_j)$$
$$=P_r(r_0)=0.19$$
$$s_1=T(r_1)=\sum_{j=0}^{1}P_r(r_j)$$
$$=P_r(r_0)+P_r(r_1)=0.44$$
$$s_2=T(r_2)=\sum_{j=0}^{2}P_r(r_j)$$
$$=P_r(r_0)+P_r(r_1)+P_r(r_2)$$
$$=0.19+0.25+0.21=0.65$$
$$s_3=T(r_3)=\sum_{j=0}^{3}P_r(r_j)$$
$$=P_r(r_0)+P_r(r_1)+P_r(r_2)+P_r(r_3)=0.81$$

以此类推，

$$s_4=0.89$$
$$s_5=0.95$$
$$s_6=0.98$$
$$s_7=1.00$$

变换函数如图 4-6(b)所示。

这里对图像只取 8 个等间隔的灰度级，变换后的 s 值也只能选择最靠近的一个灰度级的值。因此，对上述之计算值加以修正：

$$s_0\approx\frac{1}{7}\qquad s_1\approx\frac{3}{7}$$
$$s_2\approx\frac{5}{7}\qquad s_3\approx\frac{6}{7}$$

$$s_4 \approx \frac{6}{7} \qquad s_6 \approx 1$$

$$s_5 \approx 1 \qquad s_7 \approx 1$$

由上述数值可见，新图像将只有 5 个不同的灰度级别，可以重新定义一个符号：

$$s'_0 = \frac{1}{7} \qquad s'_1 = \frac{3}{7}$$

$$s'_2 = \frac{5}{7} \qquad s'_3 = \frac{6}{7}$$

$$s'_4 = 1$$

因为 $r_0 = 0$ 经变换得 $s_0 = \frac{1}{7}$，所以有 790 个像素取 s_0 这个灰度值，r_1 映射到 $s_1 = \frac{3}{7}$，所以有 1023 个像素取 $s_1 = \frac{3}{7}$ 这一灰度值，以此类推，有 850 个像素取 $s_2 = \frac{5}{7}$ 这一灰度值。但是，因为 r_3 和 r_4 均映射到 $s_3 = \frac{6}{7}$ 这一灰度级，所以有 656+329=985 个像素取这个值。同样，有 245+122+81=448 个像素取 $s_4 = 1$ 这个新灰度值。用 $n = 4096$ 来除上述这些 n_k 值便可得到新的直方图。新直方图如图 1-6(c)所示。

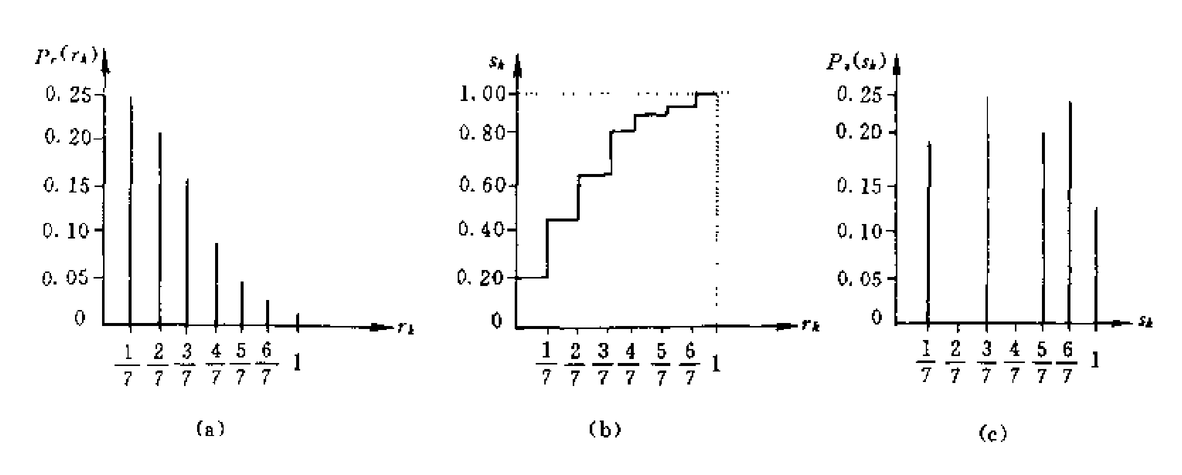


图 1-6 直方图均衡化处理

由上面的例子可见，利用累积分布函数作为灰度变换函数，经变换后得到的新灰度的直方图虽然不很平坦，但毕竟比原始图像的直方图平坦得多，而且其动态范围也大大地扩展了。因此这种方法对于对比度较弱的图像进行处理是很有效的。

因为直方图是近似的概率密度函数，所以用离散灰度级作变换时很少能得到完全平坦的结果。另外，从上例中可以看出变换后的灰度级减少了，这种现象叫做“简并”现象。由于简并现象的存在，处理后的灰度级总是要减少的。这是像素灰度有限的必然结果。由于上述原因，数字图像的直方图均衡只是近似的。

那么如何减少简并现象呢？产生简并现象的根源是利用变换公式 $s_k = \sum_{j=0}^{k} P_r(r_j)$ 求新灰度时，所得到的 s_k 往往不是允许的灰度值，这时就要采用舍入的方法求近似值，以便用与它最接近的允许灰度来代替它。在舍入的过程中，一些相邻的 s_k 值变成了相同的 s_l 值，这就发生了简并现象，于是也就造成了一些灰度层次的损失。减少简并现象的简单方法是增加像素的比特数。比如，通常用 8bit 来代表一个像素，而现在用 12bit 来表示一个像素，这样就可减少简并现象发生的机会，从而减少灰度层次的损失。另外，采用灰度间隔放大理论的直方图修正法也可以减少简并现象。这种灰度间隔放大可以按照眼睛的对比度灵敏度特性和成像系统的动态范

围进行放大。一般实现方法采用如下几步：

1）统计原始图像的直方图；

2）根据给定的成像系统的最大动态范围和原始图像的灰度级来确定处理后的灰度级间隔；

3）根据求得的步长来求变换后的新灰度；

4）用处理后的新灰度代替处理前的灰度。

以上两种方法都可以提高直方图均衡化处理的质量，大大减少由于简并现象而带来的灰度级丢失。

4.1.4 直方图规定化处理

直方图均衡化处理方法是行之有效的增强方法之一，但是由于它的变换函数采用的是累积分布函数，因此，正如前面所证明的那样，它只能产生近似均匀的直方图这样一种结果。这样就必须会限制它的效能。也就是说，在不同的情况下，并不是总需要具有均匀直方图的图像，有时需要具有特定的直方图的图像，以便能够对图像中的某些灰度级加以增强。直方图规定化方法就是针对上述思想提出来的一种直方图修正增强方法。下面仍然从研究连续灰度级的概率密度函数入手来讨论直方图规定化的基本思想。

假设 $p_r(r)$是原始图像灰度分布的概率密度函数，$p_z(z)$是希望得到的图像的概率密度函数。如何建立 $p_r(r)$和 $p_z(z)$之间的联系是直方图规定化处理的关键。

首先对原始图像进行直方图均衡化处理，即

$$s = T(r) = \int_0^r p_r(\omega)\mathrm{d}\omega \tag{4-16}$$

假定已经得到了所希望的图像，并且它的概率密度函数是 $p_z(z)$。对这幅图像也作均衡化处理，即

$$u = G(z) = \int_0^z p_z(\omega)\mathrm{d}\omega \tag{4-17}$$

因为对于两幅图像（注意，这两幅图像只是灰度分布概率密度不同）同样做了均衡化处理，所以 $p_s(s)$和 $p_u(u)$具有同样的均匀密度。其中式(4-17)的逆过程为

$$z = G^{-1}(u) \tag{4-18}$$

这样，如果用从原始图像中得到的均匀灰度级 s 来代替逆过程中的 u，其结果灰度级将是所要求的概率密度函数 $p_z(z)$的灰度级。

$$z = G^{-1}(u) = G^{-1}(s) \tag{4-19}$$

根据以上思路，可以总结出直接直方图规定化增强处理的步骤如下：

1）用直方图均衡化方法将原始图像作均衡化处理；

2）规定希望的灰度概率密度函数 $p_z(z)$，并用式(4-17)求得变换函数 $G(z)$；

3）将逆变换函数 $z=G^{-1}(s)$用到步骤(1)中所得到的灰度级。

以上三步得到了原始图像的另一种处理方法。在这种处理方法中得到的新图像的灰度级具有事先规定的概率密度函数 $p_z(z)$。

直方图规定化方法中包括两个变换函数，这就是 $T(r)$和 $G^{-1}(s)$。这两个函数可以简单地组成一个函数关系。利用这个函数关系可以从原始图像产生希望的灰度分布。

将 $s = T(r) = \int_0^r p_r(\omega)\mathrm{d}\omega$ 代入式(4-19)，有

$$z = G^{-1}[T(r)] \tag{4-20}$$

式(4-20)就是用 r 来表示 z 的公式。很显然，如果 $G^{-1}[T(r)]=T(r)$ 时，这个式子就简化为直方图均衡化方法了。

这种方法在连续变量的情况下涉及到求反变换函数的解析式的问题，一般情况下较为困难。但是由于数字处理是处理离散变量，因此，可用近似的方法绕过这个问题，从而较简单地克服了这个困难。

下面通过例子来说明处理过程。例如，这里仍用 64×64 像素的图像，其灰度级仍然是 8 级。其直方图如图 4-7(a)所示，(b)是规定的直方图，(c)为变换函数，(d)为处理后的结果直方图。原始直方图和规定的直方图之数值分别列于表 4-2 和表 4-3 中，经过直方图均衡化处理后的直方图数值列于表 4-4。

表 4-2　原始直方图数据

r_k	n_k	$P_r(r_k)=\frac{n_k}{n}$
$r_0=0$	790	0.19
$r_1=\frac{1}{7}$	1023	0.25
$r_2=\frac{2}{7}$	850	0.21
$r_3=\frac{3}{7}$	656	0.16
$r_4=\frac{4}{7}$	329	0.08
$r_5=\frac{5}{7}$	245	0.06
$r_6=\frac{6}{7}$	122	0.03
$r_7=1$	81	0.02

表 4-3　规定的直方图数据

z_k	$P_z(z_k)$
$z_0=0$	0.00
$z_1=\frac{1}{7}$	0.00
$z_2=\frac{2}{7}$	0.00
$z_3=\frac{3}{7}$	0.15
$z_4=\frac{4}{7}$	0.20
$z_5=\frac{5}{7}$	0.30
$z_6=\frac{6}{7}$	0.20
$z_7=1$	0.15

计算步骤如下：

1）对原始图像进行直方图均衡化映射处理的数值列于表 4-4 的 n_k 栏目内。

2）利用式(4-14)计算变换函数。

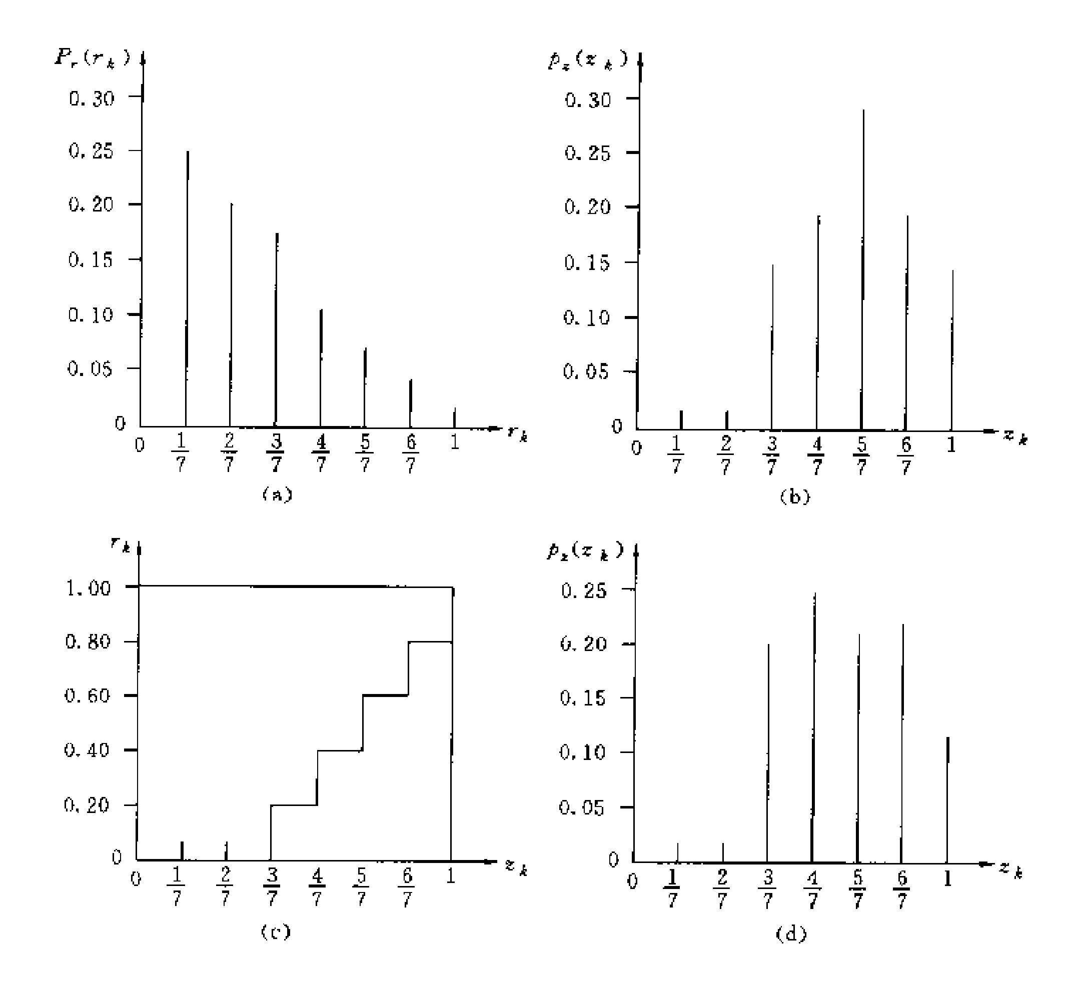


图 4-7　直方图规定化处理方法

表 4-4　均衡化处理后的直方图数据

$r_j \to s_k$	n_k	$P_s(s_k)$
$r_1 \to s_0 = \frac{1}{7}$	790	0.19
$r_1 \to s_1 = \frac{3}{7}$	1023	0.25
$r_2 \to s_2 = \frac{5}{7}$	850	0.21
$r_3, r_4 \to s_3 = \frac{6}{7}$	985	0.24
$r_5, r_6, r_7 \to s_4 = 1$	448	0.11

$$u_k = G(z_k) = \sum_{j=0}^{k} P_z(z_j)$$

$$u_0 = G(z_0) = \sum_{j=0}^{0} p_z(z_j) = p_z(z_0) = 0.00$$

$$u_1 = G(z_1) = \sum_{j=0}^{1} p_z(z_j) = p_z(z_0) + p_z(z_1) = 0.00$$

$$u_2 = G(z_2) = \sum_{j=0}^{2} p_z(z_j)$$

$$= p_z(z_0) + p_z(z_1) + p_z(z_2) = 0.00$$

$$u_3 = G(z_3) = \sum_{j=0}^{3} p_z(z_j)$$
$$= p_z(z_0) + p_z(z_1) + p_z(z_2) + p_z(z_3) = 0.15$$

以此类推求得：

$$u_4 = G(z_4) = 0.35$$
$$u_5 = G(z_5) = 0.65$$
$$u_6 = G(z_6) = 0.85$$
$$u_7 = G(z_7) = 1$$

3）用直方图均衡化中的 s_k 进行 G 的反变换求 z。

$$z_k = G^{-1}(s_k)$$

这一步实际上是近似过程。也就是找出 s_k 与 $G(z_k)$的最接近的值。例如，$s_0 = \frac{1}{7} \approx 0.14$，与它最接近的是 $G(z_3) = 0.15$，所以可写成 $G^{-1}(0.15) = z_3$。用这样方法可得到下列变换值：

$$s_0 = \frac{1}{7} \quad \rightarrow \quad z_3 = \frac{3}{7}$$
$$s_1 = \frac{3}{7} \quad \rightarrow \quad z_4 = \frac{4}{7}$$
$$s_2 = \frac{5}{7} \quad \rightarrow \quad z_5 = \frac{5}{7}$$
$$s_3 = \frac{6}{7} \quad \rightarrow \quad z_6 = \frac{6}{7}$$
$$s_4 = 1 \quad \rightarrow \quad z_7 = 1$$

4）用 $z = G^{-1}[T(r)]$找出 r 与 z 的映射关系。

$$r_0 = 0 \quad \rightarrow \quad z_3 = \frac{3}{7}$$
$$r_1 = \frac{1}{7} \quad \rightarrow \quad z_4 = \frac{4}{7}$$
$$r_2 = \frac{2}{7} \quad \rightarrow \quad z_5 = \frac{5}{7}$$
$$r_3 = \frac{3}{7} \quad \rightarrow \quad z_6 = \frac{6}{7}$$
$$r_4 = \frac{4}{7} \quad \rightarrow \quad z_6 = \frac{6}{7}$$
$$r_5 = \frac{5}{7} \quad \rightarrow \quad z_7 = 1$$
$$r_6 = \frac{6}{7} \quad \rightarrow \quad z_7 = 1$$
$$r_7 = 1 \quad \rightarrow \quad z_7 = 1$$

5）根据这样的映射重新分配像素，并用 $n=4096$ 去除，可得到最后的直方图。

由图 4-7 可见，结果直方图并不很接近希望的形状，与直方图均衡化的情况一样，这种误差是多次近似造成的。只有在连续的情况下，求得准确的反变换函数才能得到准确的结果。在灰度级减少时，规定的和最后得到的直方图之间的误差趋向于增加。但是实际处理效果表明，尽管是一种近似的直方图也可以得到较明显的增强效果。

表 4-5 结果直方图数据

z_k	n_k	$P_z(z_k)$
$z_0=0$	0	0.00
$z_1=\frac{1}{7}$	0	0.00
$z_2=\frac{2}{7}$	0	0.00
$z_3=\frac{3}{7}$	790	0.19
$z_4=\frac{4}{7}$	1023	0.25
$z_5=\frac{5}{7}$	850	0.21
$z_6=\frac{6}{7}$	985	0.24
$z_7=1$	448	0.11

实际上，从 s 到 z 的反变换有时不是单值的。当在规定的直方图中没有填满的灰度级或在 $G^{-1}(s)$ 取最接近的可能的灰度级的过程中，就会产生这种情况，如图 4-8 所示。从图中可见，$s=\frac{3}{7}$ 既可以映射成 $z=\frac{3}{7}$，也可以映射成 $\frac{4}{7}$。处理这种情况的方法一般是指定一个灰度级，使这个灰度级尽可能与给定的灰度级相配合。

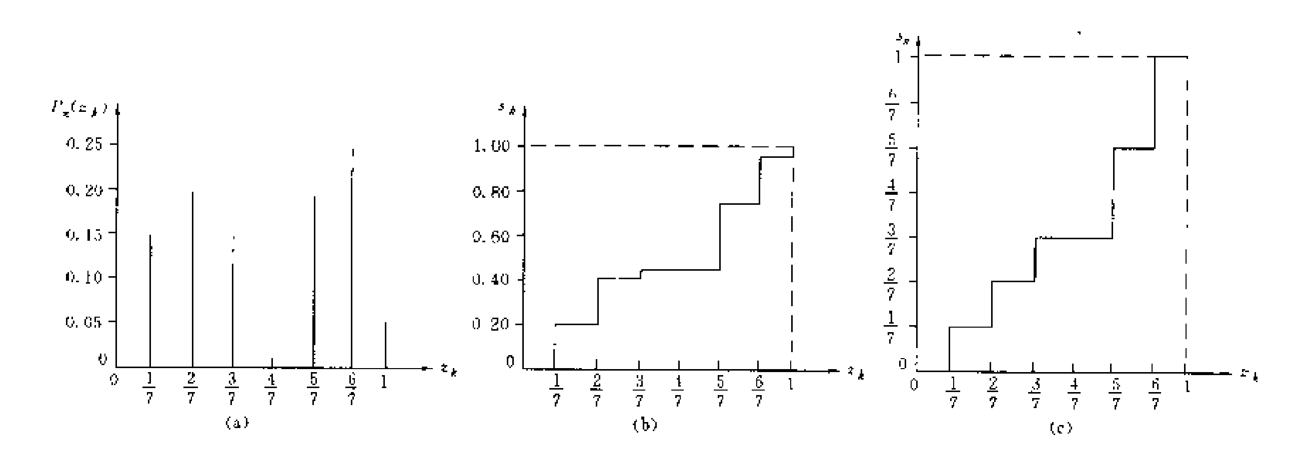


图 4-8 一种非单值变换情况

(a)规定的直方图；(b)原始变换函数；(c)舍入函数

利用直方图规定化方法进行图像增强的主要困难在于如何构成有意义的直方图。一般有两种方法，一种是给定一个规定的概率密度函数，如高斯、瑞利等函数。一些常用的直方图修正转换函数列于表 4-6 中。

表 4-6 直方图修正转换函数

规定的概率密度模型		转换函数
均匀	$P_z(z)=\frac{1}{r_{max}-r_{min}}$ $r_{min}\leqslant r\leqslant r_{max}$	$r=[r_{max}-r_{min}]p_r(r)+r_{min}$

续表

规定的概率密度模型		转换函数
指　数	$P_z(z)=a\exp\{-a(r-r_{min})\}$ $r\geqslant r_{min}$	$r=r_{min}-\frac{1}{a}\ln[1-p_r(r)]$
瑞　利	$p_z(z)=\frac{r-r_{min}}{a^2}\exp\left\{-\frac{(r-r_{min})^2}{2a^2}\right\}$ $r\geqslant r_{min}$	$r=r_{min}+\left[2a^2\ln\left\{\frac{1}{1-p_r(r)}\right\}\right]^{\frac{1}{2}}$
双曲线 （立方根）	$p_z(z)=\frac{1}{3}\frac{r^{2/3}}{r_{max}^{1/3}-r_{min}^{1/3}}$	$r=[(r_{max}^{1/3}-r_{min}^{1/3})p_r(r)+r_{min}^{1/3}]^3$
双曲线 （对数）	$p_z(z)=\frac{1}{r[\ln r_{max}-\ln r_{min}]}$	$r=r_{min}\left[\frac{r_{max}}{r_{min}}\right]p_r(r)$

另一种方法是规定一个任意可控制的直方图，其形状可由一些直线所组成，得到希望的形状后，将这个函数数字化。这种方法如图 4-9 所示。首先，在[0, 1]区间内任何地方选一点 m，并且 h 点是非负值。直方图由直线段构成，其形状受 m, h, θ_l, θ_r 4 个参量的控制，其中 θ_l 与 θ_r 是与垂线的夹角，夹角的值从 0°到 90°变化。随着 θ_l 的变化，转折点 j 沿着(0, 1)和(m, 0)两点的连线移动；同理 θ_r 变化，点 k 将沿着(1, 1)和(m, 0)的连线移动。这样，由这几条直线组成的折线可在 m, h, θ_l, θ_r 的控制下组成多种直方图。例如，当 $m=0.5$, $h=1.0$, $\theta_l=\theta_r=0$ 时就可得到一个均匀直方图。这样得到的直方图经数字化后即可做为规定的直方图使用。这种方法可以联机应用，使技术人员操纵 4 个参量，改变直方图的形状，并且连续地观察输出，并控制增强处理，使之适合于预期的目的。

4.1.5 图像对比度处理

由于图像的亮度范围不足或非线性会使图像的对比度不甚理想，可用像素幅值重新分配的方法来改善图像对比度。扩大图像的亮度范围可以用线性映射的方法，这种方法如图 4-10 所示。由图可以看出原图像的范围较小，经映射后的图像亮度范围展宽了。在这种转换中，设计转换函数应考虑到灰度量化问题，如果原始图像的灰度级为 k 级，映射后输出图像的灰度级仍然是 k 级，这样由于输出图像的灰度范围加大了，因此，使每一级灰度分层的跳变比原始图像大，由此将会产生伪轮廓效应。如果能适当地增加输出图像的灰度分层数就有可能减小这种效应。

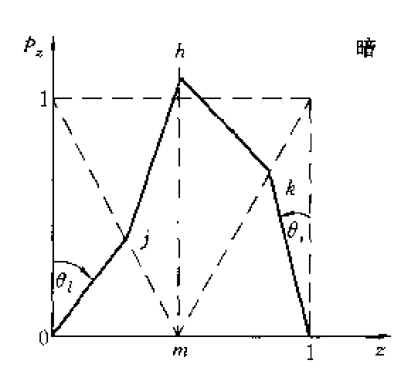


图 4-9　直方图参量规定化法

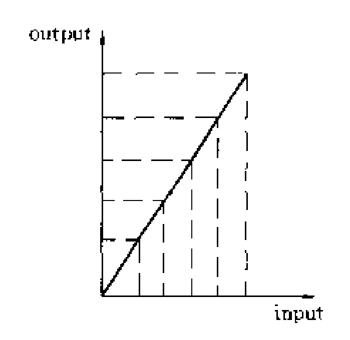


图 4-10　数字图像对比度增强

在对比度处理法中，根据不同的目的可以设计出不同的转换函数。例如图 4-10 是对比度转换函数。图 4-11 是线性转换函数，这种函数将图像在整个灰度范围内作线性映射。另外一种

映射转换函数如图 4-12 所示。这种转换是将图像中两个极端的灰度值加以限幅，这种限幅的比例也是可以选择的。除此之外，为了不同的目的还有其他一些类型的转换函数，这些转换函数的形式如图 4-13(a)、(b)、(c)所示。灰度变换的效果如图 4-14 (a)、(b) 所示，其中(a)是原像，(b)是处理后的图像。灰度反转的转换函数是把图像的低亮度区域转到较高的亮度区，而高亮度区转换为低亮度区，其效果如图 4-15 所示，其中(a)是原像，(b)是处理后的图像。锯齿形转换可以把几段较窄的输入灰度区间都扩展到整个输出灰度范围内，这种处理可以把灰度变化较平缓的区域也较鲜明地显示出来。其效果如图 4-16 所示，其中(a)是原像，(b)是处理后的图像，这里选 $n=2$。开窗式转换的目的是只对部分输入灰度区间进行转换，通过窗口位置的选择可以观察某些灰度区间的灰度分布，并且对这一区域的灰度进行映射变换。当然，图 4-13 只是举出几种常用的转换函数的形状。根据不同的需要还可以设计出更多的转换函数，其基本原理都是一样的，只不过处理效果不同罢了。经开窗式转换函数处理的图像效果如图 4-17 所示，图(a)是原像，(b)是处理后的图像。

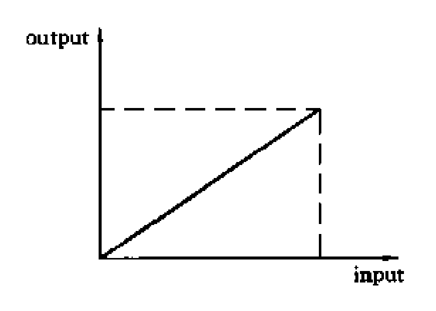


图 4-11　图像灰度的线性映射变换

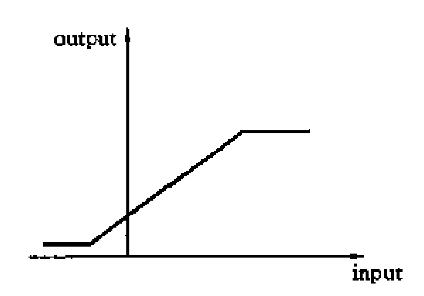


图 4-12　限幅的线性映射变换

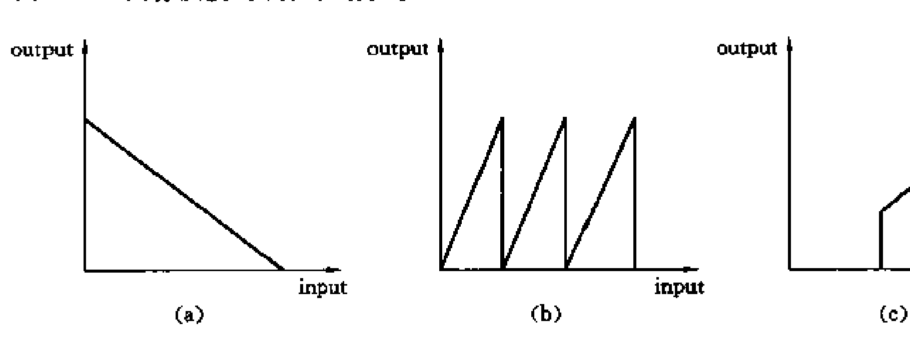
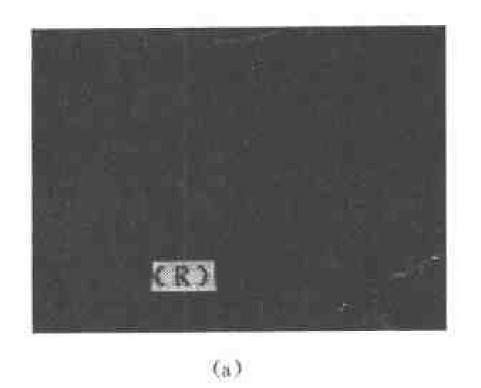
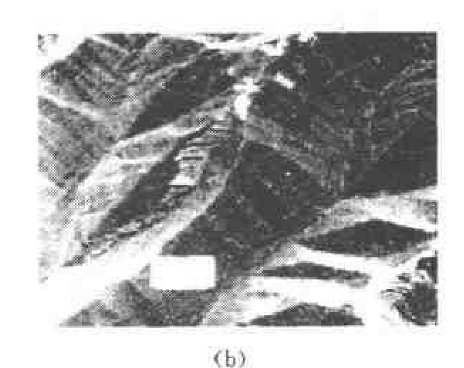


图 4-13　其他一些转换函数



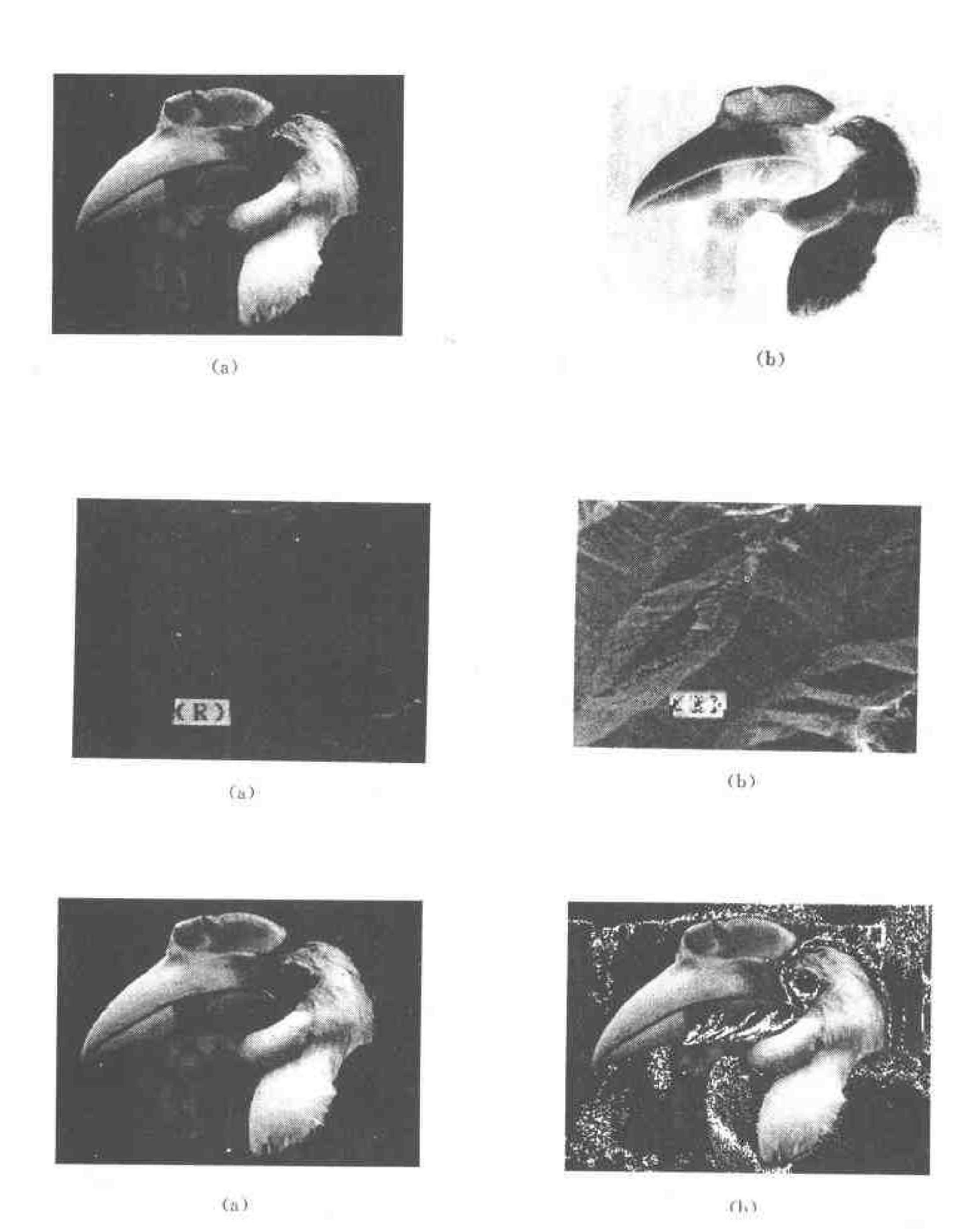
(a)



(b)

图 4-14　灰度变换处理效果

利用直方图修正技术增强图像简便而有效。直方图均衡化处理可大大改善图像灰度的动



(a) (b)

(a) (b)

(a) (b)

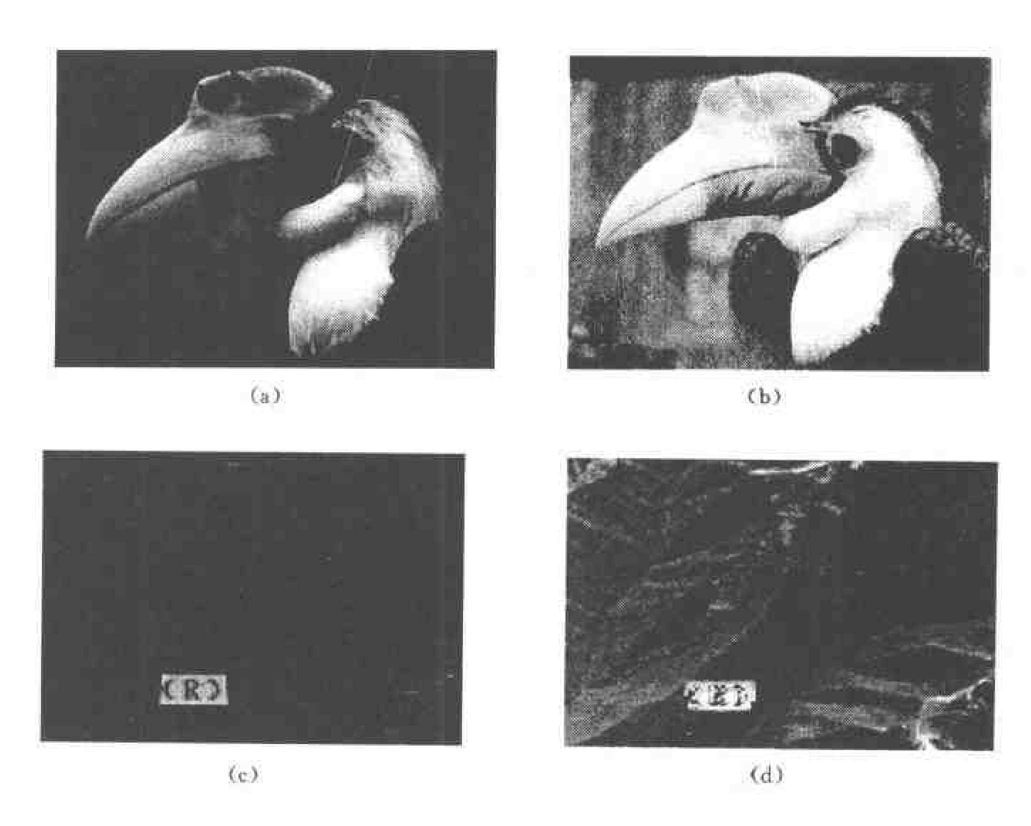


图 4-18 (a)为原始图像;(b)为均衡化处理后的图像;
(c)为原始图像;(d)为均衡化处理后的图像

附录 5 直方图均衡处理程序实例

```
// histdlg.h : header file 头文件
//

/////////////////////////////////////////////////////////////////////////////
// CHistDlg dialog

class CHistDlg : public CDialog
{
// Construction
public:
    CHistDlg(CWnd * pParent = NULL); // standard constructor

    BYTE rhist[256],ghist[256],bhist[256];
private:
    void DrawAxs();
    int color;
// Dialog Data
    //{{AFX_DATA(CHistDlg)
    enum { IDD = IDD_DIALOG6 };
```

```
        // NOTE: the ClassWizard will add data members here
    //}}AFX_DATA

// Implementation
protected:
    virtual void DoDataExchange(CDataExchange * pDX); // DDX/DDV support

    // Generated message map functions
    //{{AFX_MSG(CHistDlg)
    afx_msg void OnBlueBut();
    afx_msg void OnGreenBut();
    afx_msg void OnRedBut();
    virtual BOOL OnInitDialog();
    afx_msg void OnPaint();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

// histdlg.cpp : implementation file 处理程序
//

#include "stdafx.h"
#include "picture.h"
#include <windowsx.h>
#include "pictudoc.h"
#include "histdlg.h"
const int x=50;
const int y=265;

#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CHistDlg dialog


CHistDlg::CHistDlg(CWnd * pParent /* =NULL */)
    : CDialog(CHistDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CHistDlg)
        // NOTE: the ClassWizard will add member initialization here
    //}}AFX_DATA_INIT
```

```
void CHistDlg::DoDataExchange(CDataExchange * pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CHistDlg)
        // NOTE: the ClassWizard will add DDX and DDV calls here
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CHistDlg, CDialog)
    //{{AFX_MSG_MAP(CHistDlg)
    ON_BN_CLICKED(IDC_BLUE_BUT, OnBlueBut)
    ON_BN_CLICKED(IDC_GREEN_BUT, OnGreenBut)
    ON_BN_CLICKED(IDC_RED_BUT, OnRedBut)
    ON_WM_PAINT()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()


/////////////////////////////////////////////////////////////////////////////
// CHistDlg message handlers


void CHistDlg::OnBlueBut()
{
    // TODO: Add your control notification handler code here

    CPen * pPen=new CPen;
    pPen->CreatePen(PS_SOLID,0,RGB(0,0,255));

    CClientDC dc(this);
    dc.Rectangle(45,50,320,270);
    DrawAxs();
    CPen * pOldPen=dc.SelectObject(pPen);

    for(int i=0;i<256;i++)
    {
        dc.MoveTo(x+i,y);
        dc.LineTo(x+i,y-bhist[i]);
    }
    delete dc.SelectObject(pOldPen);
    color=3;
}
```

```
void CHistDlg::OnGreenBut()
{
    // TODO: Add your control notification handler code here

    CPen * pPen=new CPen;
    pPen->CreatePen(PS_SOLID,0,RGB(0,255,0));

    CClientDC dc(this);
    dc.Rectangle(45,50,320,270);
    DrawAxs();
    CPen * pOldPen=dc.SelectObject(pPen);

    for(int i=0;i<256;i++)
    {
        dc.MoveTo(x+i,y);
        dc.LineTo(x+i,y-ghist[i]);
    }
    delete dc.SelectObject(pOldPen);
    color=2;
}

void CHistDlg::OnRedBut()
{
    // TODO: Add your control notification handler code here

    CPen * pPen=new CPen;
    pPen->CreatePen(PS_SOLID,0,RGB(255,0,0));

    CClientDC dc(this);
    dc.Rectangle(45,50,320,270);
    DrawAxs();
    CPen * pOldPen=dc.SelectObject(pPen);

    for(int i=0;i<256;i++)
    {
        dc.MoveTo(x+i,y);
        dc.LineTo(x+i,y-rhist[i]);
    }
    delete dc.SelectObject(pOldPen);
    color=1;

}
```

```
BOOL CHistDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // TODO: Add extra initialization here
    MoveWindow(100,100,375,325,TRUE);
    color=1;
    return TRUE; // return TRUE unless you set the focus to a control
}

/////////////////////////////////////////////////////////////////////////////
void CHistDlg::DrawAxs()
{
    CClientDC dc(this);
    dc.MoveTo(x,y);
    dc.LineTo(x+260,y);          //横坐标
    dc.MoveTo(x,y);
    dc.LineTo(x,y-210);          //纵坐标
    dc.LineTo(x-3,y-210+5);
    dc.MoveTo(x,y-210);
    dc.LineTo(x+3,y-210+5); //纵坐标箭头
    dc.MoveTo(x+260,y);
    dc.LineTo(x+260-5,y-3);
    dc.MoveTo(x+260,y);
    dc.LineTo(x+260-5,y+3); //横坐标箭头
}

void CHistDlg::OnPaint()
{
    CPaintDC dc(this); // device context for painting

    // TODO: Add your message handler code here
    if(color==1) OnRedBut();
    if(color==2) OnGreenBut();
    if(color==3) OnBlueBut();
        // Do not call CDialog::OnPaint() for painting messages
}
```

4.2 图像平滑化处理

一幅图像可能存在着各种寄生效应。这些寄生效应可能在传输中产生，也可能在量化等处理过程中产生。一个较好的平滑方法应该是既能消掉这些寄生效应又不使图像的边缘轮廓和线条变模糊。这就是研究图像平滑化处理要追求的主要目标。图像平滑化处理方法有空域法

和频域法两大类。主要有邻域平均法，低通滤波法，多图像平均法等等。本节将对这些方法作一些讨论。

4.2.1 邻域平均法

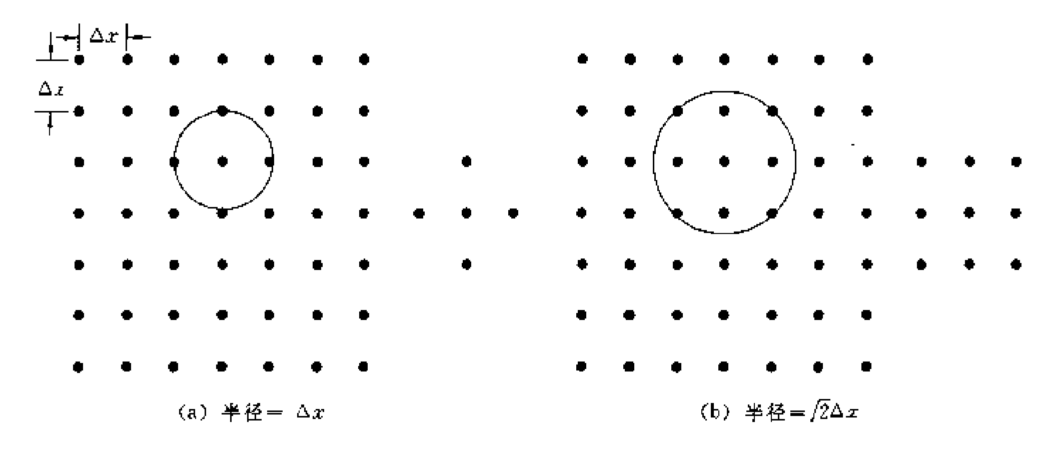
邻域平均法是简单的空域处理方法。这种方法的基本思想是用几个像素灰度的平均值来代替每个像素的灰度。假定有一幅 $N\times N$ 个像素的图像 $f(x,y)$，平滑处理后得到一幅图像 $g(x,y)$。$g(x,y)$由下式决定：

$$g(x,y)=\frac{1}{M}\sum_{(m,n)\in S}f(m,n) \tag{4-21}$$

式中 $x,y=0,1,2,\cdots,N-1$，S 是(x,y)点邻域中点的坐标的集合，但其中不包括(x,y)点，M 是集合内坐标点的总数。式(4-21)说明，平滑化的图像 $g(x,y)$中的每个像素的灰度值均由包含在(x,y)的预定邻域中的 $f(x,y)$的几个像素的灰度值的平均值来决定。例如，可以以(x,y)点为中心，取单位距离构成一个邻域，其中点的坐标集合为

$$S=\{(x,y+1),(x,y-1),(x+1,y),(x-1,y)\}$$

图 4-19 给出了两种从图像阵列中选取邻域的方法。图(a)的方法是一个点的邻域，定义为以该点为中心的一个圆的内部或边界上的点的集合。图中像素间的距离为 Δx，选取 Δx 为半径作圆，那么，点 R 的灰度值就是圆周上 4 个像素灰度值的平均值。图(b)是选 $\sqrt{2}\Delta x$ 为半径的情况下构成的点 R 的邻域，选择在圆的边界上的点和在圆内的点为 S 的集合。



(a) 半径＝ Δx　　(b) 半径＝$\sqrt{2}\Delta x$

图 4-19 在数字图像中选取邻域的方法

处理结果表明，上述选择邻域的方法对抑制噪声是有效的，但是随着邻域的加大，图像的模糊程度也愈加严重。为克服这一缺点，可以采用阈值法减少由于邻域平均所产生的模糊效应。其基本方法由下式决定：

$$g(x,y)=\begin{cases}\dfrac{1}{M}\sum\limits_{(m,n)\in S}f(m,n) & \left|f(x,y)-\dfrac{1}{M}\sum\limits_{(m,n)\in S}f(m,n)\right|>T\\ f(x,y) & \text{其他}\end{cases} \tag{4-22}$$

式中 T 就是规定的非负阈值。这个表达式的物理概念是：当一些点和它的邻域内的点的灰度的平均值的差不超过规定的阈值 T 时，就仍然保留其原灰度值不变，如果大于阈值 T 时就用它们的平均值来代替该点的灰度值。这样就可以大大减少模糊的程度。

4.2.2 低通滤波法

这种方法是一种频域处理法。在分析图像信号的频率特性时，一幅图像的边缘、跳跃部分以及颗粒噪声代表图像信号的高频分量，而大面积的背景区则代表图像信号的低频分量。用滤波的方法滤除其高频部分就能去掉噪声，使图像得到平滑。

由卷积定理可知，

$$G(u,v) = H(u,v) \cdot F(u,v) \tag{4-23}$$

其中 $F(u,v)$是含有噪声的图像的傅里叶变换，$G(u,v)$ 是平滑处理后的图像之傅里叶变换，$H(u,v)$是传递函数。选择传递函数 $H(u,v)$，利用 $H(u,v)$使 $F(u,v)$的高频分量得到衰减，得到 $G(u,v)$后再经反傅里叶变换就可以得到所希望的平滑图像 $g(x,y)$。根据前面的分析，显然 $H(u,v)$应该具有低通滤波特性，所以这种方法被称之为低通滤波法平滑化处理。低通滤波平滑化处理流程如图 4-20 所示。

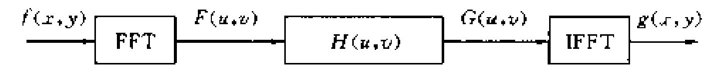


图 4-20 线性滤波器处理框图

常用的低通滤波器有下面几种。

1. 理想低通滤波器

一个理想的二维低通滤波器的传递函数由下式表示：

$$H(u,v) = \begin{cases} 1 & D(u,v) \leqslant D_0 \\ 0 & D(u,v) > D_0 \end{cases} \tag{4-24}$$

式中 D_0 是一个规定的非负的量，叫做理想低通滤波器的截止频率。$D(u,v)$是从频率域的原点到(u,v)点的距离，即

$$D(u,v) = [u^2 + v^2]^{\frac{1}{2}} \tag{4-25}$$

$H(u,v)$对 u,v 来说是一幅三维图形。$H(u,v)$的剖面图如图 4-21 所示。将剖面图绕纵轴旋转 360°就可以得到整个滤波器的传递函数。所谓理想低通滤波器，是指以截频 D_0 为半径的圆内的所有频率都能无损地通过，而在截频之外的频率分量完全被衰减。理想低通滤波器可以用计算机模拟实现，但却不能用电子元器件来实现。

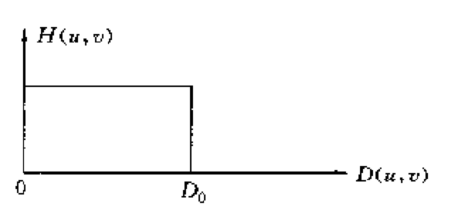


图 4-21 理想低通滤波器传递函数径向剖面图

理想低通滤波器平滑处理的概念是清晰的，但在处理过程中会产生较严重的模糊和振铃现象。这种现象正是由于傅里叶变换的性质决定的。因为滤波过程是由式(4-23)描述的，由卷积定理可知在空域中则是一种卷积关系，即

$$g(x,y) = h(x,y) * f(x,y) \tag{4-26}$$

式中 $g(x,y)$，$h(x,y)$，$f(x,y)$分别是 $G(u,v)$，$H(u,v)$，$F(u,v)$的傅里叶反变换。既然 $H(u,v)$是理想的矩形特性，那么它的反变换 $h(x,y)$的特性必然会产生无限的振铃特性。经与 $f(x,y)$卷积后则给 $g(x,y)$带来模糊和振铃现象，D_0 越小这种现象越严重，当然，其平滑效果也就较差。这是理想低通不可克服的弱点。

2. 布特沃斯(Butterworth)低通滤波器

一个 n 阶布特沃斯低通滤波器的传递函数由下式表示：

$$H(u,v) = \frac{1}{1+\left[\frac{D(u,v)}{D_0}\right]^{2n}} \tag{4-27}$$

式中 D_0 为截止频率，$D(u,v)$的值由下式决定：

$$D(u,v) = [u^2 + v^2]^{\frac{1}{2}} \tag{4-28}$$

布特沃斯低通滤波器又称最大平坦滤波器。它与理想低通滤波器不同，它的通带与阻带之间没有明显的不连续性。也就是说，在通带和阻带之间有一个平滑的过渡带。通常把 $H(u,v)$ 下降到某一值的那一点定为截止频率 D_0。在式(4-27)中是把 $H(u,v)$ 下降到原来值的 1/2 时的 $D(u,v)$定为截频点 D_0。一般情况下常常采用下降到 $H(u,v)$最大值的 $1/\sqrt{2}$ 那一点为截频点。这样，式(4-27)可修改为式(4-29)的形式：

$$H(u,v) = \frac{1}{1+(\sqrt{2}-1)\left[\frac{D(u,v)}{D_0}\right]^{2n}} \tag{4-29}$$

布特沃斯低通滤波器 $H(u,v)$的剖面图如图 4-22 所示。与理想低通滤波器的处理结果相比，经布特沃斯滤波器处理过的图像模糊程度会大大减少。因为它的 $H(u,v)$不是陡峭的截止特性，它的尾部会包含有大量的高频成份。另外，经布特沃斯低通滤波器处理的图像将不会有振铃现象。这是由于在滤波器的通带和阻带之间有一平滑过渡的缘故。另外，由于图像信号本身的特性，在卷积过程中的折迭误差也可以忽略掉。由此可知，布特沃斯低通滤波器的处理结果比理想滤波器为好。

3. 指数低通滤波器

在图像处理中常用的另一种平滑滤波器是指数低通滤波器。它的传递函数如下式表示：

$$H(u,v) = e^{-\left[\frac{D(u,v)}{D_0}\right]^n} \tag{4-30}$$

式中 D_0 为截频，$D(u,v)$由下式决定：

$$D(u,v) = [u^2 + v^2]^{\frac{1}{2}} \tag{4-31}$$

式中的 n 是决定衰减率的系数。从式(4-30)可见，如果 $D(u,v)=D_0$，则

$$H(u,v) = \frac{1}{e} \tag{4-32}$$

如果仍然把截止频率定在 $H(u,v)$最大值的$\frac{1}{\sqrt{2}}$处，那么，公式可作如下修改：

$$\begin{aligned} H(u,v) &= e^{\left[\ln\frac{1}{2}\right]\left[\frac{D(u,v)}{D_0}\right]^n} \\ &= e^{-0.347\left[\frac{D(u,v)}{D_0}\right]^n} \end{aligned} \tag{4-33}$$

指数低通滤波器传递函数的剖面图如图 4-23 所示。由于指数低通滤波器有更快的衰减率，所以，经指数低通滤波的图像比布特沃斯低通滤波器处理的图像稍模糊一些。由于指数低通滤波器的传递函数也有较平滑的过渡带，所以图像中也没有振铃现象。

4. 梯形低通滤波器

梯形低通滤波器传递函数的形状介于理想低通滤波器和具有平滑过渡带的低通滤波器之间。它的传递函数由下式表示：

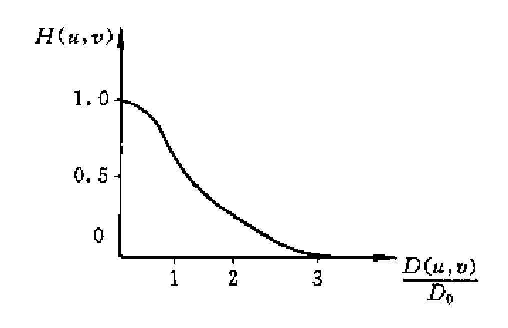


图 4-22 布特沃斯低通滤波器剖面图

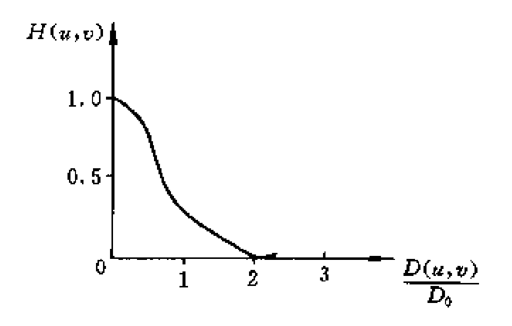


图 4-23 指数低通滤波器传递函数径向剖面图

$$H(u,v)=\begin{cases}1 & D(u,v)<D_0\\ \dfrac{1}{[D_0-D_1]}[D(u,v)-D_1] & D_0\leqslant D(u,v)\leqslant D_1\\ 0 & D(u,v)>D_1\end{cases}\tag{4-34}$$

其中 $D(u,v)=[u^2+v^2]^{\frac{1}{2}}$，在规定 D_0 和 D_1 时要满足 $D_0<D_1$ 的条件。一般为了方便，把传递函数的第一个转折点 D_0 定义为截止频率；第二个变量 D_1 可以任意选取只要 D_1 大于 D_0 就可以。梯形低通滤波器传递函数的剖面如图 4-24 所示。由于梯形滤波器的传递函数特性介于理想低通滤波器和具有平滑过渡带滤波器之间，所以其处理效果也介于其两者中间。梯形滤波法的结果有一定的振铃现象。

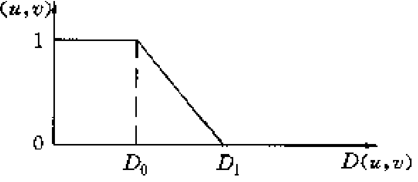


图 4-24 梯形低通滤波器传递函数剖面图

用低通滤波器进行平滑处理可以使噪声伪轮廓等寄生效应减低到不显眼的程度，但是由于低通滤波器对噪声等寄生成份滤除的同时，对有用高频成份也滤除，因此，这种去噪的美化处理是以牺牲清晰度为代价而换取的。

4.2.3 多图像平均法

如果一幅图像包含有加性噪声，这些噪声对于每个坐标点是不相关的，并且其平均值为零，在这种情况下就可能采用多图像平均法来达到去掉噪声的目的。

设 $g(x,y)$ 为有噪声图像，$n(x,y)$ 为噪声，$f(x,y)$ 为原始图像，可用下式表示：

$$g(x,y)=f(x,y)+n(x,y)\tag{4-35}$$

多图像平均法是把一系列有噪声的图像 $\{g_j(x,y)\}$ 迭加起来，然后再取平均值以达到平滑的目的。具体做法如下：

取 M 幅内容相同但含有不同噪声的图像，将它们迭加起来，然后作平均计算，如下式所示：

$$\bar{g}(x,y)=\frac{1}{M}\sum_{j=1}^{M}g_j(x,y)\tag{4-36}$$

由此得出：

$$E\{\bar{g}(x,y)\}=f(x,y)\tag{4-37}$$

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{M}\sigma^2_{n(x,y)} \tag{4-38}$$

式中 $E\{\bar{g}(x,y)\}$ 是 $\bar{g}(x,y)$ 的数学期望，$\sigma^2_{\bar{g}(x,y)}$ 和 $\sigma^2_{n(x,y)}$ 是 $\bar{g}$ 和 n 在 (x,y) 坐标上的方差。在平均图像中任一点的均方差可由下式得到：

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}}\sigma_{n(x,y)} \tag{4-39}$$

由上二式可见，M 增加则像素值的方差就减小，这说明由于平均的结果使得由噪声造成的像素灰度值的偏差变小。从式(4-37)中可以看出，当作平均处理的噪声图像数目增加时，其统计平均值就越接近原始无噪声图像。这种方法在实际应用中的最大困难在于把多幅图像配准起来，以便使相应的像素能正确地对应排列。

图 4-25 示出了图像平滑处理的效果，其中(a)是待处理图像，(b)是处理后的图像。


(a)


(b)

图 4-25 图像平滑处理效果

4.3 图像尖锐化处理

图像尖锐化处理主要用于增强图像的边缘及灰度跳变部分。通常所讲的勾边增强方法就是图像尖锐化处理。与图像平滑化处理一样，图像尖锐化处理同样也有空域和频域两种处理方法。

4.3.1 微分尖锐化处理

在图像平滑化处理中，主要的空域处理法是采用邻域平均法，这种方法类似于积分过程，积分的结果使图像的边缘变得模糊了。积分既然使图像细节变模糊，那么，微分就会产生相反的效应。因此，微分法是图像尖锐化方法之一。

微分尖锐化的处理方法最常用的是梯度法。由场论理论知道，数量场的梯度是这样定义的：

设一数量场 u，$u=u(x,y,z)$，把大小是在某一点方向导数的最大值，方向是取得方向导数最大值的方向的矢量叫数量场的梯度。由这个定义出发，如果给定一个函数 $f(x,y)$，在坐标 (x,y) 上的梯度可定义为一个矢量，

$$\mathrm{grad}[f(x,y)] = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{4-40}$$

由梯度的定义可知它有两个特点：

1）矢量 $\mathrm{grad}[f(x,y)]$是指向 $f(x,y)$最大增加率的方向；

2）如果用 $G[f(x,y)]$来表示 $\mathrm{grad}[f(x,y)]$的幅度，那么

$$\begin{aligned} G[f(x,y)] &= \max\{\mathrm{grad}[f(x,y)]\} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}} \end{aligned} \tag{4-41}$$

这就是说，$G[f(x,y)]$等于在 $\mathrm{grad}[f(x,y)]$的方向上每单位距离 $f(x,y)$的最大增加率。显然，式(4-41)是一个标量函数，并且 $G[f(x,y)]$永远是正值。由于我们经常用到的是式(4-41)，因此，在后续讨论中将笼统地称“梯度的模”为梯度。

在数字图像处理中，仍然要采用离散形式，为此用差分运算代替微分运算。式(4-41)可用下面的差分公式来近似：

$$\begin{aligned} G[f(x,y)] \approx \{&[f(x,y) - f(x+1,y)]^2 \\ &+ [f(x,y) - f(x,y+1)]^2\}^{\frac{1}{2}} \end{aligned} \tag{4-42}$$

在用计算机计算梯度时，通常用绝对值运算代替式(4-42)，所以，有式(4-43)所示的近似公式：

$$G[f(x,y)] \approx |f(x,y) - f(x+1,y)| + |f(x,y) - f(x,y+1)| \tag{4-43}$$

图 4-26 示出了式(4-43)中像素间的关系。应该注意到，对一幅 $N\times N$ 个像素的图像计算梯度时，对图像的最后一行，或者最后一列不能用式(4-43)来求解，解决方法是对这个区域的像素在 $x=N$，$y=N$ 时重复前一行和前一列的梯度值。

关于梯度处理的另一种方法是所谓的罗伯特梯度(Robert gradient)法。这是一种交叉差分法。其近似计算值如下式：

$$\begin{aligned} G[f(x,y)] \approx \{&[f(x,y) - f(x+1,y+1)]^2 \\ &+ [f(x+1,y) - f(x,y+1)]^2\}^{\frac{1}{2}} \end{aligned} \tag{4-44}$$

用绝对值近似计算式如下：

$$G[f(x,y)] \approx |f(x,y) - f(x+1,y+1)| + |f(x+1,y) - f(x,y+1)| \tag{4-45}$$

式(4-44)和(4-45)式中像素间的关系如图 4-27 所示。

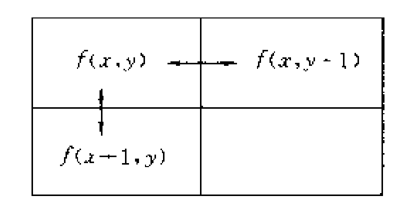


图 4-26 计算二维梯度的一种方法

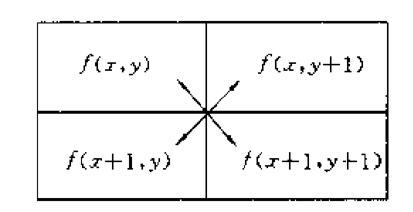
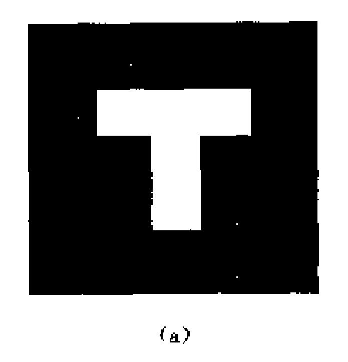


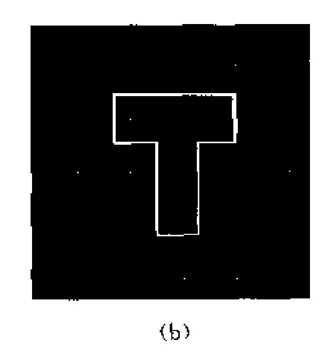
图 4-27 罗伯特梯度法

由上面的公式可见，梯度的近似值都和相邻像素的灰度差成正比。这正象所希望的那样，在一幅图像中，边缘区梯度值较大，平滑区梯度值较小，对于灰度级为常数的区域梯度值为零。这种性质正如图 4-28 所示。图中(a)是一幅二值图像，(b)为计算梯度后的图像。由于梯度运算

的结果，使得图像中不变的白区变为零灰度值，黑区仍为零灰度值，只留下了灰度值急剧变化的边沿处的点。



(a)



(b)

图 4-28 二值图像及计算梯度的结果

当选定了近似梯度计算方法后，可以有多种方法产生梯度图像 $g(x,y)$。最简单的方法是让坐标 (x,y) 处的值等于该点的梯度，即

$$g(x,y) = G[f(x,y)] \tag{4-46}$$

这个简单方法的缺点是使 $f(x,y)$ 中所有平滑区域在 $g(x,y)$ 中变成暗区，因为平滑区内各点梯度很小。为克服这一缺点可采用阈值法（或叫门限法）。其方法如下式表示：

$$g(x,y) = \begin{cases} G[f(x,y)] & G[f(x,y)] \geqslant T \\ f(x,y) & \text{其他} \end{cases} \tag{4-47}$$

也就是说，事先设定一个非负的门限值 T，当梯度值大于或等于 T 时，则这一点就取其梯度值作为灰度值，如果梯度值小于 T 时则仍保留原 $f(x,y)$ 值。这样，通过合理地选择 T 值，就有可能既不破坏平滑区域的灰度值又能有效地强调了图像的边缘。

基于上述思路的另一种作法是给边缘处的像素值规定一个特定的灰度级 L_G，即

$$g(x,y) = \begin{cases} L_G & G[f(x,y)] \geqslant T \\ f(x,y) & \text{其他} \end{cases} \tag{4-48}$$

这种处理会使图像边缘的增强效果更加明显。

当只研究图像边缘灰度级变化时，要求不受背景的影响，则用下式来构成梯度图像：

$$g(x,y) = \begin{cases} G[f(x,y)] & G[f(x,y)] \geqslant T \\ L_B & \text{其他} \end{cases} \tag{4-49}$$

式中 L_B 是规定的背景灰度值。

另外，如果只对边缘的位置感兴趣，则可采用下式的规定产生图像：

$$g(x,y) = \begin{cases} L_G & G[f(x,y)] \geqslant T \\ L_B & \text{其他} \end{cases} \tag{4-50}$$

计算方法框图如图 4-29 所示。

一种典型的边缘增强图像如图 4-30 所示，图(a)为原像，(b)为提取边缘后的图像。

图 4-30 示出了图像尖锐化处理的典型例子，图中(a)是原像，(b)是 Soble 算子处理的结果，(c)是拉普拉斯算子处理结果，(d)是个向异性处理结果。

4.3.2 零交叉边缘检测

Marr 和 Hildreth 提出的拉普拉斯边缘检测算子 $\nabla^2 G$ 被誉为最佳边缘检测器之一。在 Marr 的视觉理论中，拉普拉斯高斯算子 $\nabla^2 G$ 扮演着相当重要的角色。该算子的特点是利用高

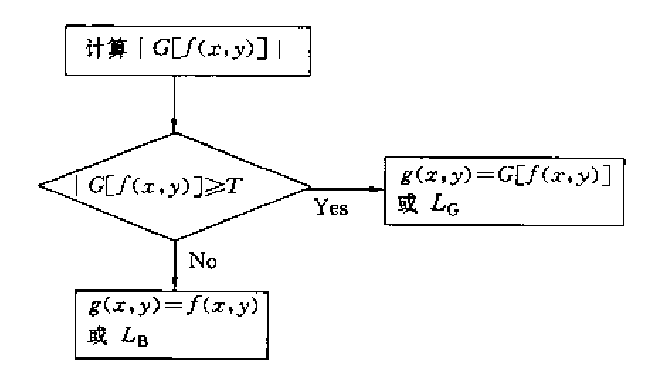


图 4-29 梯度法尖锐化处理计算框图

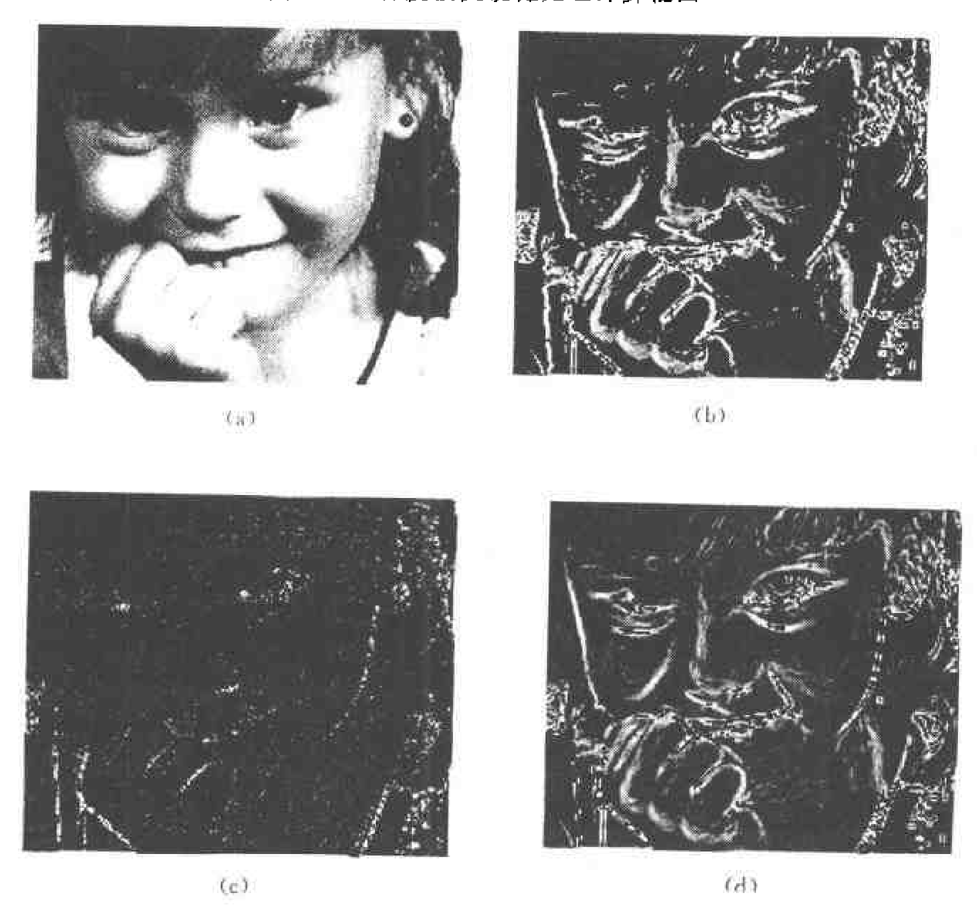


图 4-30 图像尖锐化处理的例子

斯滤波器对图像进行平滑。二维高斯滤波器的响应函数为

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4-51}$$

设 $I(x,y)$ 为灰度图像函数，由线性系统中卷积和微分的可交换性，得

$$\nabla^2\{G(x,y)*I(x,y)\}=\{\nabla^2G(x,y)\}*I(x,y) \tag{4-52}$$

即：对图像的高斯平滑滤波与拉普拉斯微分运算可结合成一个卷积算子如下：

$$\begin{aligned}\nabla^2G(x,y)&=\frac{1}{2\pi\sigma^4}\left(\frac{x^2+y^2}{\sigma^2}-2\right)e^{-\frac{x^2+y^2}{2\sigma^2}}\\&=A^2\left(\frac{x^2}{\sigma^2}-1\right)e^{-\frac{x^2}{2\sigma^2}}e^{-\frac{y^2}{2\sigma^2}}+A^2\left(\frac{y^2}{\sigma^2}-1\right)e^{-\frac{x^2}{2\sigma^2}}e^{-\frac{y^2}{2\sigma^2}}\\&=K_1(x)K_2(y)+K_1(y)K_2(x)\end{aligned} \tag{4-53}$$

$$
\left.\begin{aligned}
A &= \frac{1}{\sqrt{2\pi\sigma^2}} \\
K_1(x) &= A\left(\frac{x^2}{\sigma^2} - 1\right) e^{-\frac{x^2}{2\sigma^2}} \\
K_2(x) &= Ae^{-\frac{y^2}{2\sigma^2}}
\end{aligned}\right\} \tag{4-54}
$$

用上述算子卷积图像，通过判断符号的变化所确定出零交叉点的位置，就是边缘点。此方法又称 LOG 算法(Laplacian-of-Gaussian Algorithm)，利用$\nabla^2 G$的可分解性，对图像的二维卷积可简化为两个一维卷积如下：

$$
\begin{aligned}
\nabla^2 G * I(x,y) &= \sum_{i=-W}^{W}\sum_{j=-W}^{W} I(x-j,y-i)[(K_1(i)K_2(j) + K_2(i)K_1(j)]\sum_{j=-W}^{W} \\
&= \left\{\left[\sum_{i=-W}^{W} I(x-j,y-i)K_2(i)\right]K_1(j) + \left[\sum_{i=-W}^{W} I(x-j,y-i)K_1(i)\right]K_2(j)\right\} \\
&= \sum_{j=-W}^{W}[C(x-j,y)K_1(j) + D(x-j,y)K_2(j)]
\end{aligned}
$$

其中，

$$
\left.\begin{aligned}
C(x-j,y) &= \sum_{i=-W}^{W} I(x-j,y-i)K_2(i) \\
D(x-j,y) &= \sum_{i=-W}^{W} I(x-j,y-i)K_1(i)
\end{aligned}\right\} \tag{4-55}
$$

用零交叉算子和 Soble 算子对图像处理结果如图 4-31 所示。

4.3.3 高通滤波法

因为图像中的边缘及急剧变化部分与高频分量有关，所以当利用高通滤波器衰减图像号中的低频分量时就会相对地强调其高频分量，从而加强了图像中的边缘及急剧变化部分，达到图像尖锐化的目的。与低通滤波器相对应，常用的高通滤波器有理想高通滤波器、布特沃斯高通滤波器、指数高通滤波器和梯形高通滤波器等。这里只讨论径向对称的零相移滤波器。

1. 理想高通滤波器

一个理想的二维高通滤波器的传递函数由下式表示：

$$
H(u,v) = \begin{cases} 0 & D(u,v) \leqslant D_0 \\ 1 & D(u,v) > D_0 \end{cases} \tag{4-56}
$$

式中D_0是从频率平面原点算起的截止频率(或距离)，$D(u,v)$仍然由下式决定：

$$
D(u,v) = (u^2 + v^2)^{\frac{1}{2}} \tag{4-57}
$$

理想高通滤波器传递函数的剖面图如图 4-32 所示。

由图可见，理想高通传递函数与理想低通正好相反。通过高通滤波正好把以D_0为半径的圆内的频率成份衰减掉，对圆外的频率成份则无损地通过。与理想低通一样，理想高通可以用计算机模拟实现，但不可能用电子元件来实现。

2. 布特沃斯高通滤波器

截止频率为D_0的n阶布特沃斯高通滤波器的传递函数如下式表示：

$$
H(u,v) = \frac{1}{1 + \left[\frac{D_0}{D(u,v)}\right]^{2n}}
$$



Lena 原始图像



用 Soble 算子处理后的图像



用零交叉算子处理后的图像
尺度空间常数为 1.414



用零交叉算子处理后的图像
尺度空间常数为 1.35

图 4-31　零交叉边缘提取及与 Soble 算子法结果比较

式中，

$$D(u,v) = (u^2 + v^2)^{\frac{1}{2}} \tag{4-58}$$

布特沃斯高通滤波器传递函数的径向剖面图如图 4-33 所示

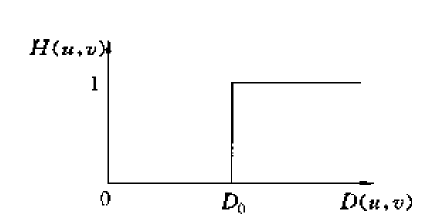


图 4-32　理想高通滤波器传递函数径向剖面图

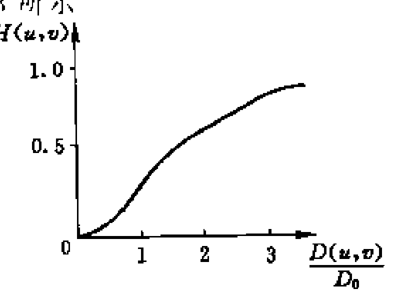


图 4-33　布特沃斯高通滤波器传递函径向剖面图($n=1$)

与低通滤波器一样，定义 $H(u,v)$ 下降到其最大值的 1/2 处的 $D(u,v)$ 为截频点 D_0。一般情况下，高通滤波器的截频选择在使 $H(u,v)$ 下降到其最大值的 $\frac{1}{\sqrt{2}}$ 处，满足这一条件的传递函数可修改成下式：

$$H(u,v) = \frac{1}{1 + [\sqrt{2} - 1]\left[\frac{D_0}{D(u,v)}\right]^{2n}} = \frac{1}{1 + 0.414\left[\frac{D_0}{D(u,v)}\right]^{2n}} \tag{4-59}$$

3. 指数高通滤波器

截频为 D_0 的指数高通滤波器的传递函数如下式表示：

$$H(u,v)=e^{-\left[\frac{D_0}{D(u,v)}\right]^n} \tag{4-60}$$

式中 D_0 为截频，$D(u,v)=(u^2+v^2)^{\frac{1}{2}}$，参数 n 控制着 $H(u,v)$ 的增长率。指数高通滤波器的传递函数径向剖面图如图 4-34 所示。

由式(4-60)可知，当 $D(u,v)=D_0$ 时，$H(u,v)=1/e$。如果仍然把截止频率定在 $H(u,v)$ 最大值的 $1/\sqrt{2}$ 时，则其传递函数可修改为下面的形式：

$$\begin{aligned} H(u,v) &= e^{\ln\frac{1}{\sqrt{2}}\left[\frac{D_0}{D(u,v)}\right]^n} \\ &= e^{-0.347\left[\frac{D_0}{D(u,v)}\right]^n} \end{aligned} \tag{4-61}$$

4. 梯形高通滤波器

梯形高通滤波器的传递函数用下式表示：

$$H(u,v)=\begin{cases} 0 & D(u,v)<D_0 \\ \dfrac{[D(u,v)-D_1]}{[D_0-D_1]} & D_1\leqslant D(u,v)\leqslant D_0 \\ 1 & D(u,v)>D_0 \end{cases} \tag{4-62}$$

同样，式中 $D(u,v)=(u^2+v^2)^{\frac{1}{2}}$。$D_0$ 和 D_1 为规定值，并且 $D_0>D_1$，定义截频为 D_0，D_1 是任选的，只要满足 $D_0>D_1$ 就可以了。梯形高通滤波器的传递函数径向剖面图如图 4-35 所示。

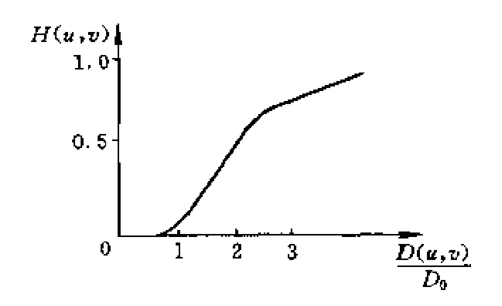


图 4-34　指数高通滤波器传递函数径向剖面图

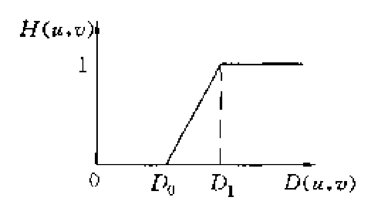


图 4-35　梯形高通滤波器传递函数径向剖面图

在图像尖锐化处理中也可以采用空域离散卷积的方法，这种方法与高通滤波相比有类似的效果。这种方法是首先确定掩模，然后作卷积处理。式(4-63)、式(4-64)和式(4-65)列出了几种掩模，式中的冲激响应阵列是高通形式的：

$$h=\begin{vmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{vmatrix} \tag{4-63}$$

$$h=\begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix} \tag{4-64}$$

$$h=\begin{vmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{vmatrix} \tag{4-65}$$

另外，用于勾边处理的另一种方法是统计差值法。这种方法是将图像中的每一像素值除以测量的统计均方差 $\sigma(x,y)$，即

$$g(x,y)=\frac{f(x,y)}{\sigma(x,y)} \tag{4-66}$$

其中方差由下式表示：

$$\sigma^2(x,y)=\sum_x\sum_y[f(x,y)-\overline{f}(x,y)]^2 \qquad x,y\in N(x,y) \tag{4-67}$$

$\sigma^2(x,y)$是在(x,y)点某一领域 $N(x,y)$内计算的。$\overline{f}(x,y)$是原始图像在(x,y)点上的平均值。

以上介绍的是图像尖锐化处理的几种方法。值得注意的是在尖锐化处理过程中，图像的边缘细节得到了加强，但图像中的噪声也同时被加重了，所以在实际处理中往往采用几种方法处理以便能得到更加满意的效果。

4.4 利用同态系统进行增强处理

利用同态系统进行图像增强处理是把频率过滤和灰度变换结合起来的一种处理方法。它是把图像的照明反射模型作为频域处理的基础，利用压缩亮度范围和增强对比度来改善图像的一种处理技术。

一幅图像 $f(x,y)$可以用它的照明分量 $i(x,y)$及反射分量 $r(x,y)$来表示，即

$$f(x,y)=i(x,y)\cdot r(x,y) \tag{4-68}$$

因为傅里叶变换是线性变换，所以对于式(1-68)中具有相乘关系的两个分量无法分开。也就是说，

$$\mathscr{F}\{f(x,y)\}\neq\mathscr{F}\{i(x,y)\}\cdot\{\mathscr{F}(r(x,y)\}$$

式中 $\mathscr{F}$ 代表傅里叶变换。如果首先把式(4-68)的两边取对数就可以把式中的乘性分量变成加性分量，而后再加以进一步处理，即

$$z(x,y)=\ln f(x,y)=\ln i(x,y)+\ln r(x,y) \tag{4-69}$$

此后，对式(4-69)两端再进行傅里叶变换，得

$$\begin{aligned}\mathscr{F}\{z(x,y)\}&=\mathscr{F}\{\ln f(x,y)\}\\&=\mathscr{F}\{\ln i(x,y)\}+\mathscr{F}\{\ln r(x,y)\}\end{aligned} \tag{4-70}$$

令

$$Z(u,v)=\mathscr{F}\{z(x,y)\}$$

$$I(u,v)=\mathscr{F}\{\ln i(x,y)\}$$

$$R(u,v)=\mathscr{F}\{\ln r(x,y)\}$$

则

$$Z(u,v)=I(u,v)+R(u,v) \tag{4-71}$$

如果用一个传递函数为 $H(u,v)$的滤波器来处理 $Z(u,v)$，那么，如前面所讨论的那样，有：

$$\begin{aligned}S(u,v)&=H(u,v)\cdot Z(u,v)\\&=H(u,v)\cdot I(u,v)+H(u,v)\cdot R(u,v)\end{aligned} \tag{4-72}$$

处理后，将式(4-72)再施以傅里叶反变换，则

$$\begin{aligned}s(x,y)&=\mathscr{F}^{-1}\{S(u,v)\}\\&=\mathscr{F}^{-1}\{H(u,v)\cdot I(u,v)\}+\mathscr{F}^{-1}\{H(u,v)\cdot R(u,v)\}\end{aligned} \tag{4-73}$$

令

$$i'(x,y) = \mathscr{F}^{-1}\{H(u,v) \cdot I(u,v)\}$$
$$r'(x,y) = \mathscr{F}^{-1}\{H(u,v) \cdot R(u,v)\}$$

式(4-73)可写成下式：

$$s(x,y) = i'(x,y) + r'(x,y) \tag{4-74}$$

因为 $z(x,y)$是 $f(x,y)$的对数，为了得到所要求的增强图像 $g(x,y)$还要进行一次相反的运算，即

$$\begin{aligned} g(x,y) &= \exp\{s(x,y)\} \\ &= \exp\{i'(x,y) + r'(x,y)\} \\ &= \exp\{i'(x,y)\} \cdot \exp\{r'(x,y)\} \end{aligned} \tag{4-75}$$

令

$$i_0(x,y) = \exp\{i'(x,y)\}$$
$$r_0(x,y) = \exp\{r'(x,y)\}$$

则

$$g(x,y) = i_0(x,y) \cdot r_0(x,y) \tag{4-76}$$

式中 $i_0(x,y)$是处理后的照射分量，$r_0(x,y)$是处理后的反射分量。

一幅图像的照射分量通常用慢变化来表征，而反射分量则倾向急剧变化。这个特征使人们有可能把一幅图像取对数后的傅里叶变换的低频分量和照射分量联系起来，而把反射分量与高频分量联系起来。这样的近似虽然是粗糙的，但是却可以收到有效的增强效果。

用同态滤波方法进行增强处理的流程框图如图 4-36 所示。

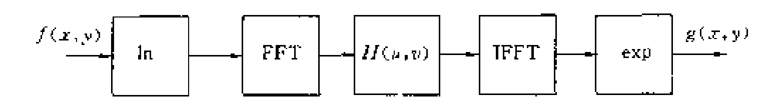


图 4-36　同态滤法增强处理流程框图

一般情况下，照明决定了图像中像素灰度的动态范围，而对比度是图像中某些内容反射特性的函数。用同态滤波器可以理想地控制这些分量。适当地选择滤波器传递函数将会对傅里叶变换中的低频分量和高频分量产生不同的响应。处理结果会使像素灰度的动态范围或图像对比度得到增强。

当处理一幅由于照射光不均匀而产生黑斑暗影时(摄像机常常会有这种缺陷)，想要去掉这些暗影又不失去图像的某些细节，则这种处理是很有效的。

4.5　彩色图像处理

前面几节讨论的都是对单色图像的处理技术。为了更有效地增强图像，在数字图像处理中广泛应用了彩色处理技术。

在图像处理中色彩的运用主要出于以下两个因素：首先，在自动图像分析中色彩是一个有力的描绘子，它通常可使从一个场景中识别和抽取目标的处理得到简化；第二，人们对图像进行分析时，人眼能区别的灰度层次大约只有二十几种，但却能够识别成千上万的色彩。

彩色图像处理被划分为三个主要领域，即：全彩色(或真彩色)、假彩色和伪彩色处理。在全

彩色处理中，被处理的图像一般从全彩色传感器中获得，例如彩色摄影机或彩色扫描仪；假彩色处理是一种尽量逼近真实彩色的人工彩色处理技术；伪彩色处理的问题是分配彩色给某种灰度（强度或强度范围），以增强辨识能力。这种20世纪80年代取得重大进步的图像处理技术由于彩色传感器和彩色图像处理的硬件的成熟而使得彩色图像处理技术得到广泛应用。

一般情况下，把能真实反映自然物体本来颜色的图像叫真彩色图像。例如，由彩色摄像机摄制，并由彩色监视器复原的彩色图像就近于真彩色。在计算机图像处理系统中进行真彩色图像处理的方法如图4-37所示。

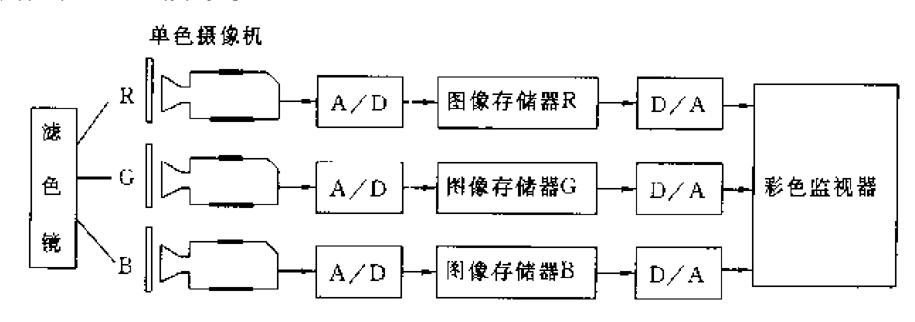


图4-37　真彩色图像处理框图

在处理过程中，首先用加有红色滤色片的摄像机（黑白摄像机）摄取彩色图像，图像信号经数字化送入一块图像存储板存起来，第二步用带有绿色滤色片的摄像机摄取图像，图像信号经数字化送入第二块图像存储板，最后用带有蓝色滤色片的摄像机摄取图像，图像数据存储在第三块图像存储板内。三幅图像数据准备好后就可以在系统的输出设备——彩色监视器上合成一幅真彩色图像。另外，利用彩色摄像机摄取彩色图像，然后利用解码电路解出红、绿、蓝三幅单色图像进行处理也是常用的彩色图像处理方法。其原理如图4-38所示。

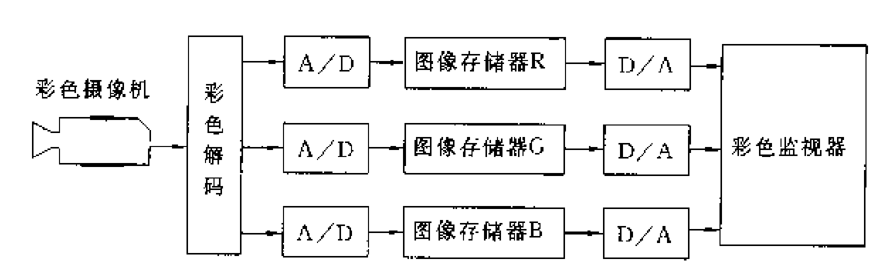


图4-38　另一种真彩色图像处理框图

关于假彩色图像处理与伪彩色图像处理这两个术语在有些文献中并没有加以严格区分。但从图像处理的角度来看，二者还是有差别的。例如，人们常常把没有颜色的人物照片用人工着色的方法彩色化，其目的主要是在于使其颜色尽量接近于真实色彩，这种技术笔者认为应归入假彩色技术。普拉特提出的假彩色概念是这样的：将一幅由三基色描绘的彩色原图像或具有同一内容的一套多光谱图像，逐像素映射到由三激励值所确定的色度空间上去，这种映射可以是线性的也可以是非线性的。

伪彩色技术早期在遥感图片处理中已有应用，采用的方法是光学方法。这种方法固然有几何失真小的优点，但其处理速度极慢，而且处理一幅照片要较复杂的洗印技术，这样有时会限制它的应用范围。一般情况下人为设计的各个目标物的颜色是不同的。如蓝色的天空可以映射为红色，绿色的草地也可以映射为蓝色等等。其主要目的在于使处理后的图像中的某些内容更加醒目，当然彩色的设计与人的视觉心理特性有很大关系。伪彩色技术也可以应用于线性彩

色坐标变换，它可以由原图像基色转变为另一组新基色。

伪彩色数字图像处理是电处理方法。它可以实时处理，而且其精度可以做得很高。在处理结果需要保留时，可有多种方法制做硬拷贝。为什么伪彩色处理可以达到增强的效果呢？这主要是由于人眼对彩色的分辨能力远远大于对黑白灰度的分辨率。对于一般的观察者来说，通常只能分辨十几级灰度。就是经过专门训练的人员（如 X 射线透视医生）也只能分辨几十级灰度。而对于彩色来说，人的眼睛可分辨出上千种彩色的色调和强度。因此，在一幅黑白图像中检测不到的信息，经伪彩色增强后可较容易地被检测出来。

伪彩色处理也是一种彩色映射过程。主要目的是增强观察者对图像信息的检测能力。它的主要方法是将二维数据阵列转化到色平面上。这种映射可由式(4-77)表示：

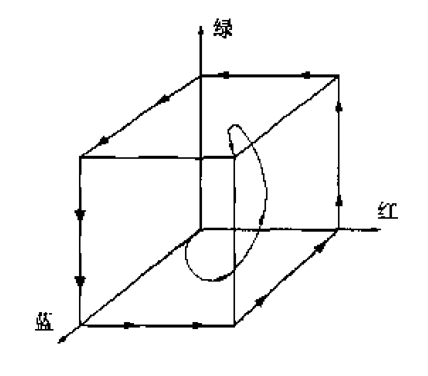


图 4-39　伪彩色映射轨迹

$$\left.\begin{aligned} R(x,y) &= \mathscr{P}_R\{f(x,y)\} \\ G(x,y) &= \mathscr{P}_G\{f(x,y)\} \\ B(x,y) &= \mathscr{P}_B\{f(x,y)\} \end{aligned}\right\} \tag{4-77}$$

式中 $R(x,y)$、$G(x,y)$，$B(x,y)$是彩色显示三激励值，$\mathscr{P}_R$，$\mathscr{P}_G$，$\mathscr{P}_B$ 是映射算子，它们可以是线性的也可以是非线性的。

通过以上映射关系就可以确定出三维色空间的轨迹。这种映射关系如图 4-39 所示。当在色度空间里给定了一条伪彩色轨迹后还要选择数据面上变量和轨迹上距离增量之间的标度关系，一般将轨迹长度分为相等的增量距离。

4.5.1　颜色基本原理

尽管在获得色彩时人脑所进行的处理是一个仍未被完全理解的生理心理现象，但颜色的物理特性在实验和理论的基础上可完全地表达出来。

1666 年，牛顿发现当一束太阳光穿过一个玻璃棱镜时，光束的边缘并不是白色的，而是一个连续的光谱，一端是紫色，另一端是红色。这条彩色光谱可分为 6 个宽域：紫色、蓝色、绿色、黄色、桔黄和红色。当用全彩色法观看，彩带中并没有哪种截然的颜色界限，而是每种彩条都平滑地变成另外一种。

人们从一个物体上获得的颜色的感觉取决于这个物体的反射光的自然特性。也就是说，可见光由电磁能谱中相对狭长的频带组成。如果物体的反射光在所有可见光波长中的成份相对平衡时，物体显示出白色。如果物体只反射有限的可见频谱范围内的某些光波时，物体则显示不同的颜色。例如，物体反射光的波长主要是从 500 到 570nm，而吸收多数其他波长的能量时，物体则显示绿色。

光的特点是颜色科学的中心。如果光线没有颜色，它唯一的属性就是它的强度或灰度。无色光通常就是我们在黑白电视机中所看到的画面，而且它已成为图像处理中所讨论的主要对象。因此，灰度一词指的是强度的数量量度，即从黑到灰最后到白。

有色光的电磁能谱范围大约从 400 到 700nm。有三个基本参数可用来描述可见光源的特性，即：辐射、灰度和亮度。辐射是通过光的能量总和，它通常用瓦特(W)衡量，灰度用流明(lm)来衡量，是观察者从光源获得的能量的量度。例如，从工作于红外线区的光谱中发射的光线可能有很大的能量，但观察者几乎感觉不到它，因为它的灰度几乎是零。亮度是一个客观标志，它是无色光强度概念的表征，而且是一个描述颜色概念的关键因素。

由于人眼的结构，所有颜色都被看作是三种所谓的基色红、绿、蓝的不同组合。为了标准化，CIE 在 1931 年给这三种基色规定了以下特定的波长值，即蓝为 435.8nm，绿为 546.1nm，红为 700nm。但从某种意义上来说，没有哪种单一的颜色可被看成为红、绿、蓝。这样为了标准化而规定三种特定颜色的波长并不意味着这三种固定的单一波长的红、绿、蓝三基色能产生所有的颜色。因为"基本"这个词的使用有可能使人们错误地认为这三种标准的基色当以不同的强度比例混合时能产生所有不同的颜色。

基色可以产生合成色光——紫红色（红加蓝）、蓝绿色（绿加蓝）和黄色（红加绿）。以一种合适的浓度混合这三种基色，或混合合成色光和与它的补色可产生白光。它说明了这三种基色和它们的组合可产生合成光。

光的基色和颜料或着色剂的基色之间是有区别的。颜料或着色剂的基色被定义为吸收一种基本有色光并反射另两种光。因此，颜料的基色是紫红色、蓝绿色和黄色，合成色是红色、绿色和蓝色。三种颜料基色的适当组合或一种合成色与它的补色的组合产生黑色。

彩色电视机的颜色是采用相加混色。许多彩色电视显像管的内部结构由大量的荧光粉组合构成。红、绿、蓝三色点一般呈三角形或一字形按组排列。当显像时，一组中的每个点都产生一种基色。如发红光的像素点的强度由显像管的电子枪来调整，其强度正比于"红色脉冲"的能量。每组中的绿色和蓝色像素点也以同样的方式来调整，显示在电视接收机上。这种效果就是来自每个像素点的三基色被加到一起并由眼中的锥状体接收，我们就感觉到了一幅全彩色图像。每秒钟刷新 25（或 30）帧就会产生连续图像。

通常用来区别一种颜色与另一种颜色的特征量有亮度、色度和饱和度。亮度用来表征颜色强度概念，色度是一种与波长有关的属性，色度表征了观察者所获得的主导颜色的感觉。当我们说一个物体是红色、桔黄或黄色时，也就确定了它的色度。与色度相比较饱和度是在颜色中搀杂白色光的数量多少的度量。纯谱颜色是完全饱和的。如紫色（红和白）和淡紫色（紫和白）这样的颜色是不饱和的。饱和的程度与添加白光的数量成比例。

色度与饱和度合在一起被称为彩色。因此，一种颜色可能是以它的亮度和彩色为特征，用来形成任何一种特定颜色的红、绿、蓝被称为三原色值，并相应地用 X，Y，Z 表示。一种颜色可以由它的三原色系数来确定，定义如下：

$$x = \frac{X}{X + Y + Z} \tag{4-78}$$

$$y = \frac{Y}{X + Y + Z} \tag{4-79}$$

$$z = \frac{Z}{X + Y + Z} \tag{4-80}$$

显然从这些等式有如下关系：

$$x + y + z = 1 \tag{4-81}$$

一种表征颜色的途径是我们已经介绍过的色度图，它以一种 x（红）和 y（绿）的函数表示颜色组成。对任意 x 和 y 的值，相应的 z（蓝）的值可从公式（4-81）中获得，即：$z=1-(x+y)$。例如，某颜色有大约 62%的绿色和 25%的红色，那么，由公式（4-81）可知，蓝色的组成大约有 13%。

各种谱色的位置（从紫色的 380nm 到红色的 780nm）标示在舌形色度图的边缘上。这些是纯谱色。任何不在边缘上而在中间的点都代表一些谱色的混合色。定位在图的边缘上的任意一点是完全饱和的。当一点越离开边缘点接近等能量点，加入的白光就越多，它就越不饱和。等

能量点的饱和度为零。

色度图对于颜色混合是非常有用的。因为图中连接任意两点的直线定义了所有不同颜色的变化，这种变化可以通过另外合并这两种颜色来获得。例如，一条从红画到绿点的直线，如果红光比绿光多，代表新颜色的点将在该连线上，但它离红点比离绿点要近。同样，一条从等能量点画到图的边缘上任意点的直线将定义一个特定的谱色。

4.5.2 颜色模型

彩色模型化的目的是按某种标准利用基色表示颜色。实质上，一种颜色模型是用一个三维坐标系统及这个系统中的一个子空间来表示。在这个系统中每种颜色都由一个单点表示。

通常使用的多数彩色模型或者是面向硬件设备(例如彩色监视器或打印机)，或者是面向应用的。在实际中，通常使用的与硬件有关的模型有 RGB 模型，这种模型用在彩色监视器和彩色摄像机等领域。GMY 模型用在彩色打印机上。YIQ 模型用于彩色电视广播。在第三种模型中，Y 相对于亮度，而 I 和 Q 是被称为正交的两个颜色分量。在所有模型中最常用于彩色图像的是 HSI 模型和 HSV 模型。

在图像处理中最常用的模型是 RGB、YIQ 和 HSI 模型。下面我们将介绍这三种模型的基本特性，并且讨论它们的不同点和在数字图像处理中的应用。尽管 CMY 模型用于打印，而不是用于实际的图像处理，我们在这里也介绍一下，因为它在获得硬件拷贝输出上很重要。

1. RGB 彩色模型

在 RGB 模型中，每种颜色的主要光谱中都有红、绿、蓝的成份。这种模型基于 Cartesian(笛卡儿)坐标系统。颜色子空间如图 4-40 的立方体所示，在图中，RGB 值在 3 个顶角上，蓝绿色、紫红色和黄色在另三个顶角上，黑色在原点，白色在离原点最远的角上。在这个模型中，灰度级沿着黑白两点的连线从黑延伸到白，其他各种颜色由位于立方体内或立方体上的点来表示，同时由原点延伸的矢量决定。为了方便，假定所有的颜色值都已被标准化，图 4-40 中的立方体就是单位立方体。也就是，所有 R、G、B 的值都被假定在[0,1]范围内。

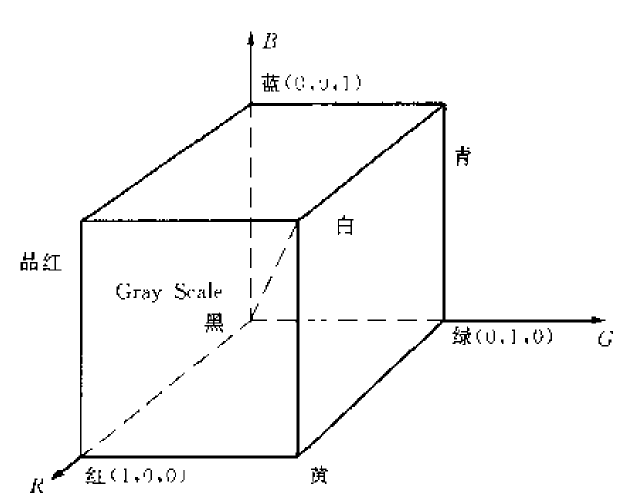


图 4-40 R、G、B 彩色矩形

RGB 彩色模型中的图像由三个独立的图像平面构成，每个平面代表一种原色。当输入 RGB 监视器时，这三个图像在屏幕上组合产生了合成的彩色图像。这样，当图像本身用 3 原色平面描述时，在图像处理中运用 RGB 模型就很有意义。相应地，大多数用来获取数字图像的彩

色摄像机都使用 RGB 格式。目前，在图像处理中只使用这种重要模型。

RGB 模型应用的一个例子是航天和卫星多光谱图像数据的处理。图像是由工作于不同光谱范围的图像传感器获得的。例如，一帧 LANDSAT 陆地卫星图像由 4 幅数字图像组成。每幅图像有相同的场景，但通过不同的光谱范围或窗口获得，两个窗口在可见光谱范围内，大致对应于绿和红，另两个窗口在光谱的红外线部分。这样每幅图像平面都有物理意义。

如果对人脸的彩色图像进行增强处理，部分图像隐藏在阴影中，直方图均衡是处理这类问题的理想工具。如果应用 RGB 模型，因为存在三种图像（红、绿、蓝），而直方图均衡仅根据强度值处理，很显然，如果把每幅图像单独地进行直方图均衡，所有可能隐藏在阴影中的图像部分都将被增强。然而，所有三种图像的强度将不同地改变颜色性能（如色调），显示在 RGB 监视器上时就不再是自然和谐的了。因此，RGB 模型对于这类处理就不太合适。

2. CMY 彩色模型

如前所述，蓝绿色、红紫色和黄色都是光的合成色（或二次色）。例如，当用白光照蓝绿色的表面时没有红光从这个表面反射出来。也就是说，蓝绿色从反射的白光中除去红光，这白光本身由等量的红绿蓝光组成。

多数在纸上堆积颜色的设备，如彩色打印机、复印机，要求 CMY 数据输入或进行一次 RGB 到 CMY 的变换。这一变换可以用一简单的变换式表示：

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4-82}$$

这里，假定所有的颜色值都已被标准化到[0,1]范围内。式(4-82)表明从一个纯蓝绿色表面反射的光线中不包括红色（即 $C=1-R$）。类似地，纯红紫色不反射绿色，纯黄色不反射蓝色。式(4-82)揭示了 RGB 值可以很容易地用 1 减 CMY 单个值的方法获得。如前所述，CMY 模型在图像处理中用在产生硬拷贝输出上，因此，从 CMY 到 RGB 的反变换操作通常没有实际意义。

3. YIQ 彩色模型

YIQ 彩色模型用于彩色电视广播。为了有效传输并与黑白电视兼容，YIQ 是一个 RGB 的编码。实际上，YIQ 系统中的 Y 分量提供了黑白电视机要求的所有影像信息。RGB 到 YIQ 的变换定义为

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & -0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4-83}$$

为了从一组 RGB 值中获得 YIQ 值，我们可简单地进行矩阵变换。YIQ 模型利用人的可视系统对亮度变化比对色调和饱和度变化更敏感而设计的。这样，YIQ 标准中用以表示 Y 时给予较大的带宽（指数字颜色所用比特数），用以表示 I、Q 时赋予较小的带宽。

另外，它成为普遍应用的标准是因为在图像处理中 YIQ 模型的主要优点是去掉了亮度(Y)和颜色信息(I 和 Q)间的紧密联系。亮度是与眼中获得的光的总量成比例的。去除这种联系的重要性在于处理图像的亮度成份时能在不影响颜色成份的情况下进行。例如，前面提到的 RGB 模型。我们可以采用直方图均衡技术对由 YIQ 格式的彩色图像进行处理，即通过给它的 Y 成份进行直方图均衡处理，图像中相关的颜色不受处理影响。

4. HSI 彩色模型

回想一下前节中讨论的色调是描述纯色（纯黄、桔黄或红）的颜色属性。而饱和度提供了由

白光冲淡纯色程度的量度。HSI 颜色模型的重要性在于两方面，第一，去掉强度成份(I)在图像中与颜色信息的联系；第二，色调和饱和度成份与人们获得颜色的方式密切相关。这些特征使 HSI 模型成为一个理想的研究图像处理运算法则的工具，这个法则基于人的视觉系统的一些颜色感觉特性。

很多实用系统都用到 HSI 模型，如：自动判断水果和蔬菜的成熟度的图像处理系统，用颜色样本匹配或检测彩色产品品质的图像处理系统等。在这些相似的应用中，关键是把系统操作建立在颜色特性上，人们用这些特性完成特定的任务。

从 RGB 到 HSI 的变换公式比以前的模型更复杂，我们花时间推得这些公式是为了给读者一个颜色处理的更深的理解。

5. 由 RGB 到 HSI 的转换

如前面所讨论的，RGB 模型的定义与单位立方体图有关。HSI 模型的颜色分量(色调(hue)和饱和度(saturation)的定义与图 4-41(a)所示的彩色三角形有关。在图 4-41(a)中，我们注意到，颜色点 P 的色调 H 是该向量与红色轴的夹角。因此，当 $H=0°$时，为红色；$H=60°$时，为黄色等等。色点 P 的饱和度 S 是指一种颜色被白色稀释的程度，它与 P 点到三角形中心的距离成正比。P 点距三角形中心越远，这种颜色的饱和度越大。

HSI 模型中的亮度的测量与垂直于三角形并通过其中心的直线有关。沿着位于三角形下方的直线，亮度逐渐由暗到黑，相反，在三角形上方，亮度逐渐由明亮变到白。

在三维色空间中将色调、饱和度、亮度结合起来，就产生了如图 4-41(b)所示的三面的、类似金字塔的结构。这个结构表面上的任意一点代表一种完全饱和的颜色。这种颜色的色调由它与红色轴的夹角决定，亮度由该点与黑色点的垂直距离决定(与黑色点的距离越远，亮度越强)。类似的结论也适用于结构内的点，唯一不同的是，随着它们逐渐接近纵轴，颜色的饱和度逐渐降低。

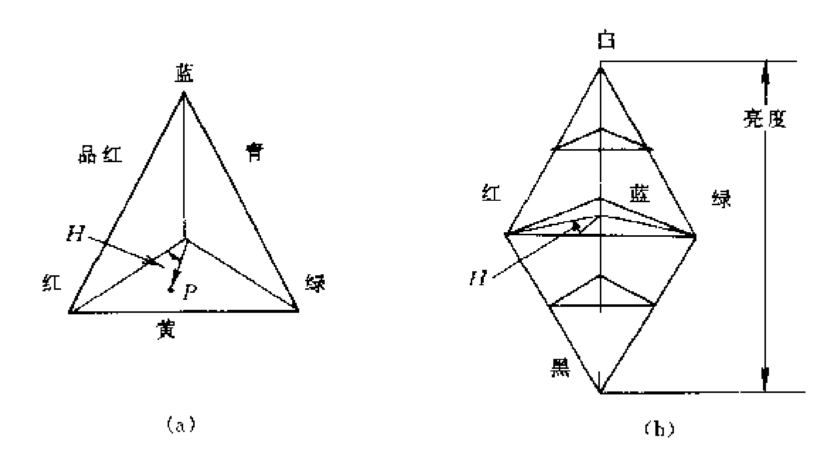


图 4-41　HIS 彩色三角形(a)和彩色立体图(b)

HSI 模型的颜色定义与归一化的红、绿、蓝值有关。这些值由 RGB 的三基色给出：

$$r=\frac{R}{(R+G+B)} \tag{4-84}$$

$$g=\frac{G}{(R+G+B)} \tag{4-85}$$

$$b=\frac{B}{(R+G+B)} \tag{4-86}$$

在此我们假定 R、G、B 已被归一化，其值在[0,1]之间。式(4-84)～式(4-86)说明 r、g、b 的值也在[0,1]之间，而且

$$r - g + b = 1 \tag{4-87}$$

我们注意到，尽管 R、G、B 可同时为 1，但归一化变量必须满足式(4-87)。事实上，式(4-87)是包含 HSI 三角形的平面的等式。

对任意三个[0,1]范围内的 R、G、B 颜色分量，HSI 模型的亮度 I 可定义为

$$I = \frac{1}{3}(R + G + B) \tag{4-88}$$

上式得出一个[0,1]范围内的值。

要得到色调 H 和饱和度 S。要得到 H 值需要有如图 4-42 所示的 HSI 三角形的几何构造，从中我们可以注意到以下条件：

1) 点 W 的坐标为$\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right)$。

2) 一个任意颜色点 P 的坐标为(r,g,b)。

3) $\boldsymbol{W}$ 表示由原点引向点 W 的向量，与此相似，$\boldsymbol{P}_R$ 和 $\boldsymbol{P}$ 分别表示由原点引向点 P_R 和 P 的向量。

4) 直线 $P_iQ_i(i=R、G、B)$在 W 点相交。

5) 使 $r_0=R/I$，$g_0=G/I$，$b_0=B/I$，其中 I 由式(4-88)给出。从图 4-42(a)中可以看出，P_RQ_R 是点(r_0,g_0,b_0)的轨迹。因为在 P_RQ_R 线上，$g_0=b_0$。与此类似，在 P_BQ_B 上，$r_0=g_0$，在 P_GQ_G 上，$r_0=b_0$。

6) 在三角形 $P_RQ_RP_G$ 所围的平面区域内的任意点有 $g_0 \geqslant b_0$。在三角形 $P_RQ_RP_B$ 所围的平面区域内的任意点有 $b_0 \geqslant g_0$。因此，线段 P_RQ_R 划分了 $g_0>b_0$ 区域和 $g_0<b_0$ 区域。类似地，P_GQ_G 划分了 $b_0<r_0$ 区域和 $b_0<r_0$ 区域，P_BQ_B 划分了 $g_0>r_0$ 区域和 $g_0<r_0$ 区域。

7) $i=R$、G 或 B 时，$|WQ_i|/|P_iQ_i|=1/3$，$|WP_i|/|P_iQ_i|=2/3$。在此，$|\arg|$代表幅度的模。

8) 定义 RG 部分是 WP_RP_G 所围区域，GB 部分是 WP_GP_B 所围区域，BR 部分是 WP_BP_R 所围区域。

参考图 4-42(a)，任意颜色的色调可用线段 WP_R 和 WP 的夹角来定义，或用向量形式(图 4-42(c))，由向量$(\boldsymbol{P}_r-\boldsymbol{W})$和$(\boldsymbol{P}-\boldsymbol{W})$的夹角来定义。例如，像前面提到过的，$H=0°$代表红，$H=120°$代表绿等等。尽管角度 H 的测量可相对于任何通过 W 的直线，然而以红色轴为基准测量色调是一个约定。总的来说，在 $0° \leqslant H \leqslant 180°$时，有下式成立：

$$(\boldsymbol{P} - \boldsymbol{W})\cdot(\boldsymbol{P}_R - \boldsymbol{W}) = \|\boldsymbol{P} - \boldsymbol{W}\| \cdot \|\boldsymbol{P}_R - \boldsymbol{W}\| \cdot \cos H \tag{4-89}$$

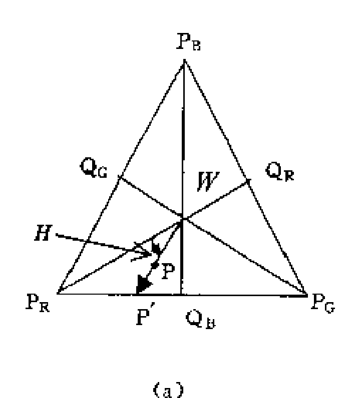
这里，$\boldsymbol{X}\cdot\boldsymbol{Y}=\boldsymbol{X}^T\boldsymbol{Y}=\|\boldsymbol{X}\|\cdot\|\boldsymbol{Y}\|\cos H$ 表示两个向量的点积或内积，两竖线代表向量的模。现在的问题是怎样以 RGB 三基色的形式来表达结果。

由条件(a)和(b)，

$$\|\boldsymbol{P} - \boldsymbol{W}\| = \left[\left(r - \frac{1}{3}\right)^2 + \left(g - \frac{1}{3}\right)^2 + \left(b - \frac{1}{3}\right)^2\right]^{\frac{1}{2}} \tag{4-90}$$

因为带有分量 a_1,a_2,a_3 的向量 $\boldsymbol{a}$ 的模为 $\|\boldsymbol{a}\|=[a_1^2+a_2^2+a_3^2]^{\frac{1}{2}}$。将式(4-84)～式(4-86)代入式(4-90)，化简后得到：

$$\|\boldsymbol{P} - \boldsymbol{W}\| = \left[\frac{9(R^2+G^2+B^2) - 3(R+G+B)^2}{9(R+G+B)^2}\right]^{\frac{1}{2}} \tag{4-91}$$



(a)

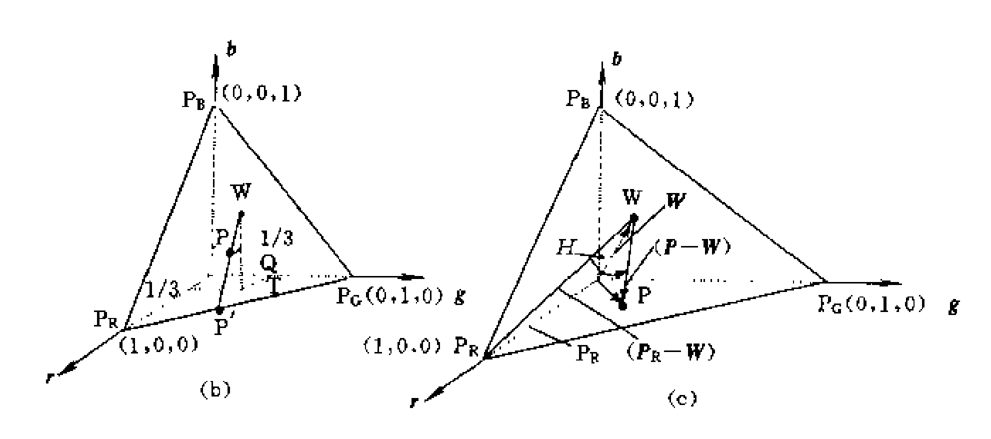


图 4-42　HIS 彩色三角形对色调及色饱和度描述的详图

当向量 $\boldsymbol{P}_{\mathrm{R}}$ 和 $\boldsymbol{W}$ 分别由原点指向点(1,0,0)和点(1/3,1/3,1/3)时，

$$\| \boldsymbol{P}_{\mathrm{R}} - \boldsymbol{W} \| = \left(\frac{2}{3} \right)^{\frac{1}{2}} \tag{4-92}$$

应记住，对向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b} = a_1b_1 + a_2b_2 + a_3b_3$，因此，

$$\begin{aligned}(\boldsymbol{P} - \boldsymbol{W})(\boldsymbol{P}_{\mathrm{R}} - \boldsymbol{W}) &= \frac{2}{3}\left(r - \frac{1}{3}\right) - \frac{1}{3}\left(g - \frac{1}{3}\right) - \frac{1}{3}\left(b - \frac{1}{3}\right) \\ &= \frac{2R - G - B}{3(R + G + B)}\end{aligned} \tag{4-93}$$

由式(4-89)，有

$$H = \arccos\left[\frac{(\boldsymbol{P} - \boldsymbol{W}) \cdot (\boldsymbol{P}_{\mathrm{R}} - \boldsymbol{W})}{\| \boldsymbol{P} - \boldsymbol{W} \| \cdot \| \boldsymbol{P}_{\mathrm{R}} - \boldsymbol{W} \|} \right] \tag{4-94}$$

将式(4-91)～(4-93)代入式(4-94)中，化简得 R、G、B 形式的 H 的表达式：

$$H = \arccos\left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\} \tag{4-95}$$

式(4-95)得到在 $0° \leqslant H \leqslant 180°$ 之间的 H 值。当 $b_0 > g_0$ 时，H 将大于 180°。因此，我们令 $H = 360° - H$。有时，色调表达式通过运用三角恒等式 $\arccos x = 90° - \arctan\left(\frac{x}{\sqrt{1 - x^2}} \right)$，以正切形式表示，但式(4-95)不仅看起来更简单，而且在硬件实现方面也更优越。

下一步我们要求出以 RGB 三基色的值表示的饱和度 S 的表达式。参考图 4-42(a)和(b)，因为颜色的饱和度是颜色被稀释的程度。由图 4-42(a)，色点 P 的饱和度 S 由比例 $|WP|/|WP'|$ 给出。延长直线 WP 与最近的三角形相交可得到 P' 点。

参考图 4-42(b)，令点 T 为 W 在 rg 平面上的投影，WT 平行于 b 轴，令 Q 为 P 在 WT 上的投影，PQ 平行于 rg 平面，那么

$$S=\frac{|WP|}{|WP'|}=\frac{|WQ|}{|WT|}=\frac{|WT|-|QT|}{|WT|} \tag{4-96}$$

这里，等式的第二步是根据相似三角形得出的。

因为 $|WT|=1/3$，$|QT|=b$，所以

$$\begin{aligned}S &= 3(1/3-b)\\ &= 1-3b\\ &= 1-b_0\end{aligned} \tag{4-97}$$

最后一步可根据式(4-87)和条件(5)导出。我们也注意到，在 RG 部分，$b_0=\min\{r_0,g_0,b_0\}$。事实上，类似的理由可说明关系式：

$$\begin{aligned}S &= 1-\min\{r_0,g_0,b_0\}\\ &= 1-\frac{3}{(R+G+B)}\cdot\min\{R,G,B\}\end{aligned} \tag{4-98}$$

对 HSI 三角形内的任意点都是普遍适用的。

为了由[0,1]范围的 RGB 值得到同样在[0,1]范围内的 HSI 值，上述结论得出了以下几个表达式：

$$I=1/3(R+G+B) \tag{4-99}$$

$$S=1-\frac{3}{(R+G+B)}[\min\{R,G,B\}] \tag{4-100}$$

$$H=\arccos\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\} \tag{4-101}$$

如前面所述，当$\left(\frac{B}{I}\right)>\left(\frac{G}{I}\right)$时，$H=360°-H$。为了将色调归一化至[0,1]范围内，令 $H=H/360°$。最后，如果 $S=0$，由(4-91)式可知，|WP|必须为 0。这意味着 W 和 P 已经变为了同一点，这使定义 H 无意义。因此，当饱和度为 0 时，色调无定义。同样，由(4-94)式，当 $I=0$ 时，饱和度无定义。

6. 由 HSI 到 RGB 的转换

已知[0,1]区间的 HSI 值，现在我们想得到同样范围内的相应的 RGB 值。分析取决于条件(8)中定义的哪一部分包含给定的 H 值。我们从令 $H=360°H$ 开始，这使色调恢复到[0,360°]的范围。

在 RG 部分($0°<H<120°$)，由式(4-97)有：

$$b=1/3(1-S) \tag{4-102}$$

注意到在图 4-38(a)中，r 值是 P 在红色轴上的投影，我们可以得到 r 值。考虑图 4-43 所示三角形 P_RQQ_R，这里，O 是 rgb 坐标系的原点，三角形的斜边是图 4-42(a)中的线段 P_RQ_R，并且，连接 O，P_R 的直线是包含 r 值的红色轴，虚线是三角形 P_ROQ_R 和包含点 P 的平面的交线，这条虚线垂直于红色轴。这两个条件意味着这个平面也包含 r 值。此外，P_RQ_R 与此平面的交点包含点 P 在 P_RQ_R 上的投影。由图 4-42(a)可知，该点为 $|WP|\cos H$。由相似三角形

$$\frac{|P_RQ_R|}{|P_RO|}=\frac{a}{d} \tag{4-103}$$

由于 $|P_RO|=1$，$d=1-r$，并且 $a=|P_RQ_R|-(|WP|\cos H+|WQ_R|)$，将这些结果代入式

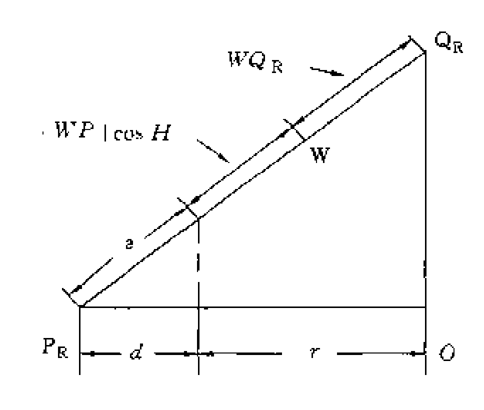


图 4-43　P_ROQ_R 三角形

(4-103)并化简得：

$$r=\frac{|WQ_R|}{|P_RQ_R|}+\frac{|WP|}{|P_RQ_R|}\cos H$$
$$=\frac{1}{3}+\frac{|WP|}{|P_RQ_R|}\cos H \tag{4-104}$$

此处，我们使用了由图 4-42(a)所得的 $|P_RQ_R|=3|WQ_R|$。上式惟一未知的是 $|WP|$。由式(4-96)，可知 $|WP|=S|WP'|$。在图 4-42(a)中，线段 P_RQ_R 和 WQ_B 在 W 点相交所得的角为 60°，因此，$|WQ_B|=|WP'|\cos(60°-H)$ 或 $|WP'|=|WQ_B'|/\cos(60°-H)$。注意到 $|WQ_B|=|WQ_R|$，将此结果代入式(4-104)可得：

$$r=\frac{1}{3}+\frac{S|WQ_R|\cos H}{|P_RQ_R|\cos(60°-H)}$$
$$=\frac{1}{3}\left[1+\frac{S\cos H}{\cos(60°-H)}\right] \tag{4-105}$$

这里，我们又用了 $|P_RQ_R|=3|WQ_R|$。最后由式 $r+g+b=1$ 可知 $g=1-(r+b)$，因此，当 $0°<H<120°$时，结论为

$$b=\frac{1}{3}(1-S) \tag{4-106}$$

$$r=\frac{1}{3}\left[1+\frac{S\cos H}{\cos(60°-H)}\right] \tag{4-107}$$

$$g=1-(r+b) \tag{4-108}$$

由式 $r+g+b=1$ 的定义，上面得到的颜色分量是归一化了的。由式(4-84)～式(4-88)，我们可以恢复 RGB 分量 $R=3I_r$，$G=3I_g$，$B=3I_b$。

在 GB 部分($120°<H<240°$)，类似的推导可得出：

$$H=H-120° \tag{4-109}$$

$$r=1/3(1-S) \tag{4-110}$$

$$g=\frac{1}{3}\left[1+\frac{S\cos H}{\cos(60°-H)}\right] \tag{4-111}$$

$$b=1-(r+g) \tag{4-112}$$

根据前面的方法，可由 r、g、b 值得到 R、G、B 值。

在 BR 部分($240°<H<360°$)，

$$H=H-240° \tag{4-113}$$

$$g=\frac{1}{3}(1-S) \tag{4-114}$$

$$b=\frac{1}{3}\left[1+\frac{S\cos H}{\cos(60°-H)}\right] \tag{4-115}$$

$$r=1-(g+b) \tag{4-116}$$

如前所述，可由 r、g、b 值得到 R、G、B 值。

4.5.3 伪彩色图像处理

1. 等密度分层伪彩色技术

等密度分层伪彩色处理是应用较多的一种方法。这种处理可以用专用硬件来实现，也可以用查表的方法来实现。密度分层是一个沿用术语，它最初来源于照相技术，因为一幅照片的浓淡层次是由照相底片上银粒的沉积度决定的，所以照片的反差(相当于电视画面的对比度)直接与密度有关。在图像处理技术中更为常用的术语是灰度一词，因此密度分层就是灰度分层。

在这一节中，我们将研究几种根据黑白图像的灰度级为之分配颜色的方法。

2. 灰度分割(Itensity slicing)

灰度分割和颜色编码是伪彩色图像处理的最简单的例子之一。如果一幅图像可被看作一个二维亮度函数，这种方法可理解为用一些平行于图像坐标平面的平面，每一平面在与函数相交处分割函数。图 4-44 展示了一个用平面 $f(x,y)=I_l$，将函数分割为两部分的例子。

如果在图 4-44 所示平面两侧分配不同颜色，那么，灰度级在平面以上的所有像素将用一种颜色编码，而灰度级在平面以下的所有像素将用另一种颜色编码，灰度级恰好位于平面本身的像素可任意分配两种颜色之一，结果是一个两色图像。将切割平面沿灰度级坐标上下移动可控制图像的外观。

多分层过程也类似图 4-44 所示的原理。具体过程可作如下解释，可作若干个平行于 xy 坐标面的平面，那么每个平面将与函数 $f(x,y)$ 相交，这种就把 $f(x,y)$ 表示的连续灰度分成若干级别，分层数可根据需要的精度加以任意设置。例如，在所定的灰度级 $L_1,L_2,\cdots,L_M$ 处定义 M 个平面，这些平面是等间距的，这 M 个平面把灰度分成 $M+1$ 个区域，区域级别为 $L_1,L_2,\cdots,L_M$，令 L_1 代表黑色$[f(x,y)=0]$，L_M 代表白色$[f(x,y)=L_M]$。那么，设 $0\leqslant M\leqslant L$，这样就完成了等密度分层。然后，可根据下面的关系式分配颜色：

$$f(x,y)=C_k \qquad f(x,y)\in R_k \tag{4-117}$$

这里，C_k 是与切割平面定义的第 k 个区间 R_k 相关的颜色。

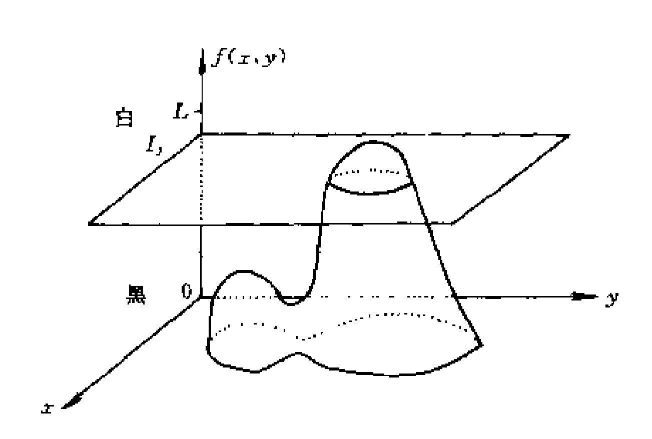


图 4-44 密度分层技术示意图

分层过程也可由图 4-45 所示的电路来实现。首先通过分压器得到一组均匀间隔的基准电压，这个基准电压送入比较器作为比较标准。图像电信号加到比较器的另一端。当信号的幅度超过比较器的基准电压时，比较器的输出端便输出一个脉冲。这样，不同的比较器的输出脉冲便代表一个不同的灰度层次，达到灰度分割的目的。

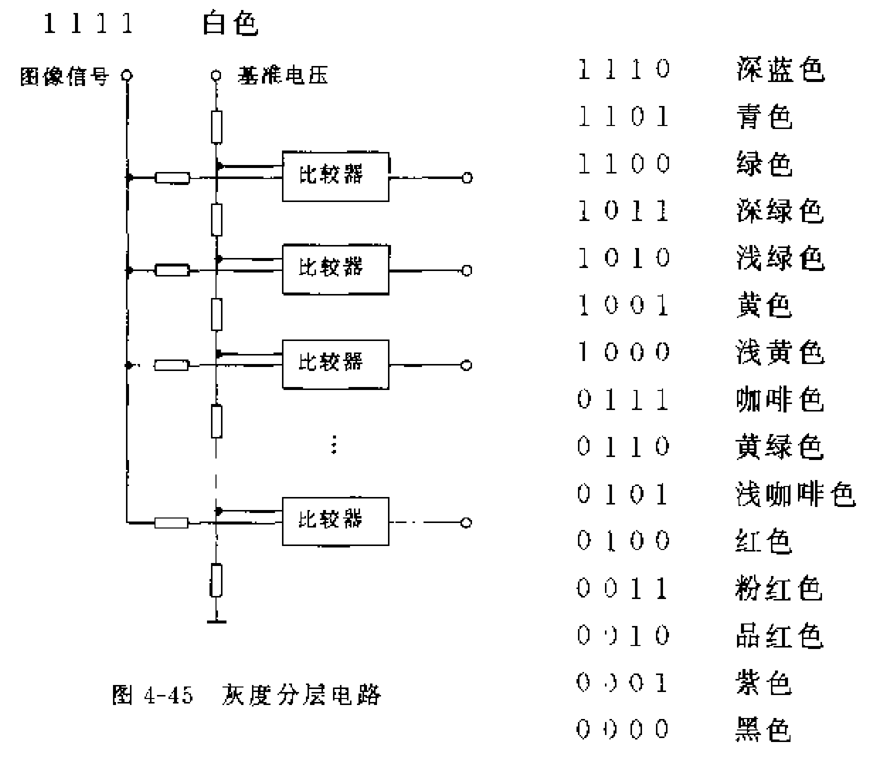
灰度-彩色变换的任务是给不同的灰度级赋于不同的颜色。它既可以结合软件用计算机来实现，也可以用硬件来实时实现。最后彩色显示都是将彩色编码送到彩色监视器的 RGB 电路上合成一幅彩色图像。

一种实时硬件编码方案如下，首先可以把灰度分割器作成 16 级分层，当然，这 16 级分层既可对整个图像信号的灰度范围，也可以行进行灰度窗口处理，对灰度整个动态范围内的某一局部范围进行分层。通过编码器可以得到四位码，将这四位码分别加到监视器中的红、绿、蓝、亮度 4 个通道中即可得到 16 种不同的色调。彩色编码原理框图如图 4-46 所示。

编码电路可以用门电路实现。

颜色与码的对应关系完全可以由人为控制来改变。例如下列所示的对应关系。

1 1 1 1　　白色



1 1 1 0　　深蓝色
1 1 0 1　　青色
1 1 0 0　　绿色
1 0 1 1　　深绿色
1 0 1 0　　浅绿色
1 0 0 1　　黄色
1 0 0 0　　浅黄色
0 1 1 1　　咖啡色
0 1 1 0　　黄绿色
0 1 0 1　　浅咖啡色
0 1 0 0　　红色
0 0 1 1　　粉红色
0 0 1 0　　品红色
0 0 0 1　　紫色
0 0 0 0　　黑色

图 4-45　灰度分层电路

当然，还可以控制加到三支枪上的信号的大小，各种色调的饱和度还可以连续调整，这样，彩色的变化就会相当丰富了。

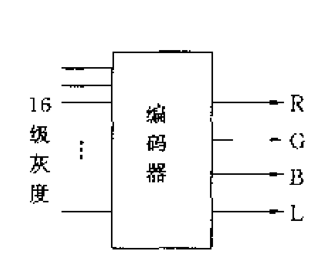


图 4-46　硬件彩色编码原理

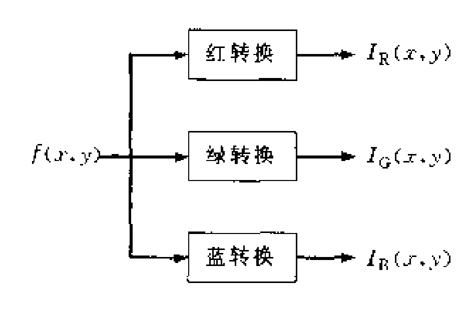


图 4-47　伪彩色图像处理框图

3. 灰度级转换为彩色

另外几种转换方法比上一节讨论的分割技术更为普遍，因而也能更好的实现伪彩色的增强效果。其中特别吸引人的一种方法如图 4-47 所示。这种方法的基本思想是对输入像素的灰度级进行三个相互独立的转换，然后，这三个结果分别送到彩色监视器的红、绿、蓝的电子发射枪上。这种方法产生了一幅彩色图像，它的颜色内容由转换函数的性质决定。注意到，这是对图像的灰度级值的转换，而不是位置的函数。

另外一种灰度-彩色变换方案可如图 4-48 所示。

这种方法的基本概念是首先对灰度动态范围开窗，然后对窗口内的灰度进行三个独立的

变换，变换的结果分别加到彩色监视器的 R、G、B 三个控制通道去，这样就可以在监视器上得到受变换函数控制的彩色合成图像。

图 4-49 示出了一种转换，如在机场中的安全检查 X 射线监测仪，左边显示的是正常物体，右边显示的可能是塑胶炸药，当选择正确的分割点时，有可能两种物品用不同的颜色区分出来。

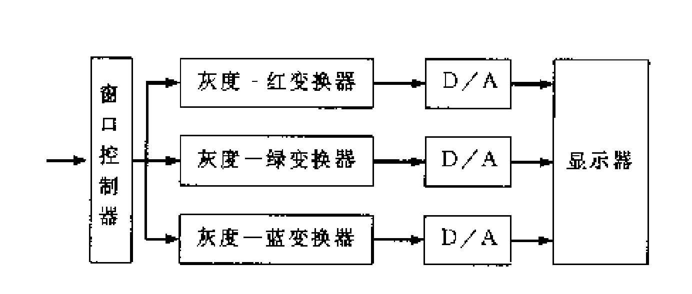


图 4-48 另一种灰度-彩色变换方案

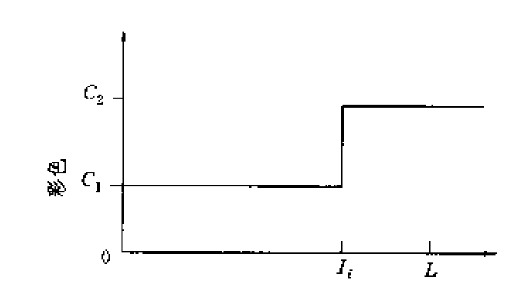


图 4-49 两种颜色的显示方法

图 4-50 示出一种转换函数。

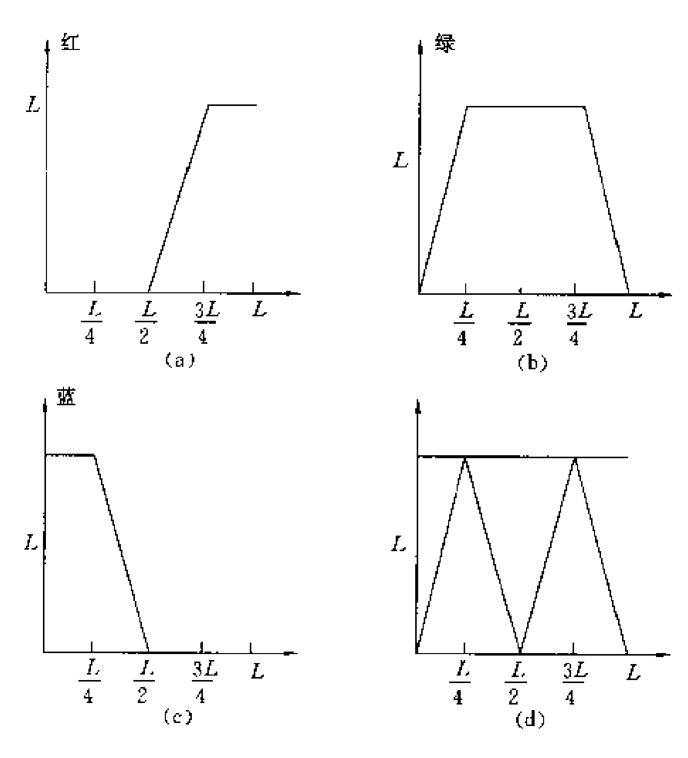


图 4-50 一组彩色变换函数

图 4-50 中的(a)是灰度-红色变换函数，在这个函数中将低于$\frac{L}{2}$的所有灰度映射成最暗的红色，在$\frac{L}{2}$到$\frac{3L}{4}$之间的灰度映射为线性增加饱和度的红色，在$\frac{3L}{4}$到 L 之间的灰度映射为最亮的红色。同样道理，绿色映射变换器如(b)所示，从 0 到$\frac{L}{4}$绿色亮度线性增加，从$\frac{L}{4}$到是$\frac{3L}{4}$最亮的绿色，从$\frac{3L}{4}$到 L 绿色亮度线性递减。蓝色映射变换函数如(c)所示，从 0 到$\frac{L}{4}$为最亮的蓝色，从$\frac{L}{4}$到$\frac{L}{2}$为线性递减特性，从$\frac{L}{2}$到 L 映射为最暗的蓝色。三种变换函数的合成特性如(d)所

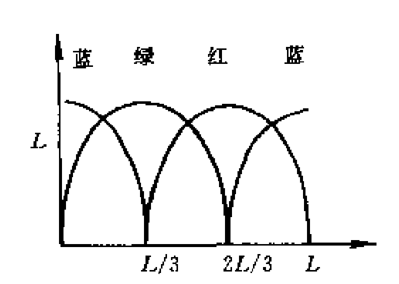


图 4-51 另一种灰度-彩色变换函数

示。从(d)中可以看到纯基色只在 0、$\frac{L}{2}$和 L 处出现,其他灰度将会合成多种不同的颜色。根据以上原理及某些特定的需要还可以设计出更多的变换函数。上述变换函数比较适合医学图像处理,而图 4-51 所示的变换函数则更适合遥感图片处理。

4. 一种滤波处理

上面介绍的是灰度-彩色变换方法。在实际应用中,根据需要也可以针对图像中的不同频率成份加以彩色增强,以便更有利于抽取频率信息。这就是基于频域运算的编码方案,原理如图 4-52 所示。这一方案与前面讨论过的基本滤波处理相同,一幅图像的傅里叶变换用三个独立的滤波函数处理以产生三个图像,以便送入彩色监视器的红、绿、蓝三个输入端。作为一个例子,下列步骤得到红色通道的图像。用一个特定的滤波函数改变输入图像的傅里叶变换,然后,再用傅里叶反变换得到处理后的图像。这些处理步骤在图像被送入监视器的红色通道之前可加入一些附加的如直方图均衡等处理。类似的处理也可以用于图 4-52 的另两个通道。这一彩色处理技术的目的是针对基于频率内容的彩色编码范畴。一个典型的滤波处理是使用低通、带通和高通滤波器以得到三个范围的频率分量。带阻和带通滤波器是已经讨论过的低通和高通滤波器概念的延伸。对于产生一个以(u_0,v_0)为原点,以环行邻域为抑制或衰减频率的滤波器的简单处理是对前边讨论过的高通滤波器进行坐标变换。对于理想滤波器其程序如下:一种理想的带阻滤波器(IBRF)抑制以(u_0,v_0)为圆心,以 D_0 为半径的邻域内的所有频率。其传递函数由下式给出:

$$H(u,v)=\begin{cases}0 & D(u,v)\leqslant D_0\\ 1 & D(u,v)>D_0\end{cases} \tag{4-118}$$

式中

$$D(u,v)=[(u-u_0)^2+(v-v_0)^2]^{\frac{1}{2}} \tag{4-119}$$

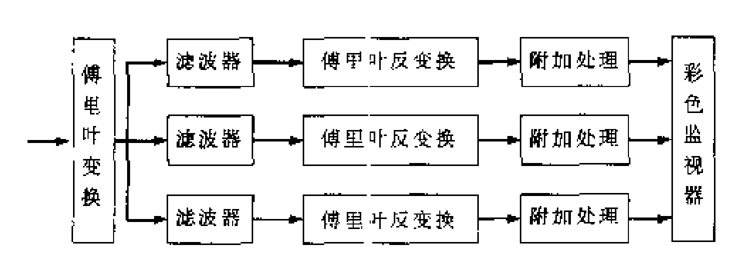


图 4-52 频率-彩色变换模型

我们注意到式(4-118)与前边讨论的滤波器是相同的,但距离函数 $D(u,v)$是以点(u_0,v_0)计算而不是以原点。由于傅里叶变换的对称性,为了得到有意义的结果,非原点的带阻滤波器必须以对称的方式进行。在理想滤波器的情况下,式(4-118)变成下式:

$$H(u,v)=\begin{cases}0 & D_1(u,v)\leqslant D_0 \quad \text{or} \quad D_2(u,v)\leqslant D_0\\ 1 & \text{其他}\end{cases} \tag{4-120}$$

这里

$$D_1(u,v)=[(u-u_0)^2+(v-v_0)^2]^{\frac{1}{2}} \tag{4-121}$$

$$D_2(u,v)=[(u+u_0)^2+(v+v_0)^2]^{\frac{1}{2}} \tag{4-122}$$

这一过程可扩展至四个或更多的区域。在前边讨论过的布特沃斯滤波器可根据我们谈到的理想滤波器技术直接推出带阻滤波器。上边讨论的滤波器是关于远离傅里叶变换原点的一

些点。为了移动以原点为中心的频带，可以考虑类似于前边讨论过的低通和高通滤波器那样的滤波器，对于理想的和布特沃斯滤波器可遵循如下步骤：一个径向对称的理想带阻滤波器关于原点的频带移动由下式给出：

$$H(u,v)=\begin{cases}1 & D(u,v)<D_0-\dfrac{W}{2}\\ 0 & D_0-\dfrac{W}{2}\leqslant D(u,v)\leqslant D_0+\dfrac{W}{2}\\ 1 & D(u,v)>D_0+\dfrac{W}{2}\end{cases} \tag{4-123}$$

这里 W 是带宽，D_0 是中心点。如果是一个径向的对称的滤波器，该滤波器可完全由一个横截面来确定。例如，一个 n 阶的径向对称的布特沃斯带阻滤波器有如下的传递函数：

$$H(u,v)=\frac{1}{1+\left[\dfrac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}} \tag{4-124}$$

这里 W 是带宽，D_0 是中心点。带通滤波器通过指定带宽内的频率，而衰减或阻止所有其他频率。因此，它们恰好是带阻滤波器的对立面。这样，如果 $H_R(u,v)$ 是带阻滤波器的传递函数，则相应的带通滤波器的传递函数可简单地用“倒转”的方法得到。即

$$H(u,v)=-[H_R(u,v)-1] \tag{4-125}$$

4.5.4 关于彩色显示

在彩色图像处理中彩色显示方法主要有两种，一种是电子显示法，也就是用彩色监视器显示，有时称这种显示为软拷贝。当然这种方法可以用彩色摄影法得到硬拷贝图像。另一种方法就是用彩色硬拷贝设备进行彩色显示。这两种设备的彩色显示原理是不同的。在日常活动中所见到的各种颜色是不同波长的光波混合而成的。任何谱色都可以用三基色按不同比例混合得到。在数字图像处理中所用的电显示法是彩色显像管，它是用相加混色法产生各种颜色的。相加混色的规律就是

红色＋绿色＝黄色
红色＋蓝色＝紫色
蓝色＋绿色＝青色
红色＋蓝色＋绿色＝白色

这里的三基色是红色、绿色、蓝色。青色、紫色、黄色称为它们的补色。

彩色硬拷贝设备是用相减混色原理显示彩色图像的。相减混色有下列规律：

黄色＝白色－蓝色
紫色＝白色－绿色
青色＝白色－红色
黄色＋紫色＝白色－蓝色－绿色＝红色
黄色＋青色＝白色－蓝色－红色＝绿色
紫色＋青色＝白色－绿色－红色＝蓝色
黄色＋紫色＋青色＝白色－蓝色－绿色－红色＝黑色

以上的关系均指颜料的混合。需要注意的是在绘画颜料方面，画家们提出的三基色是黄、蓝、红，所以绘画中的基色与色度学中的基色是不同的。

6

4.5.5 实时伪彩色增强系统

下面介绍一种实时伪彩色增强系统。这套装置实际上是在闭路电视系统中加入伪彩色处理器构成的。伪彩色处理器采用密度分割原理。伪彩色增强装置的原理框图如图 4-53 所示。

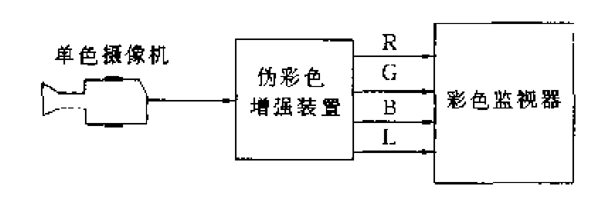


图 4-53 伪彩色增强系统框图

在这套增强仪中设置一个密度标尺，以便随时对比增强后彩色图像中不同彩色的密度值。这一标尺放在画面的上端。灰度分割及彩色编码电路原理已在前介绍过了，下面简介一下标尺发生电路的电原理图。标尺发生电路由放大器、锁相环、同步分离电路、阶梯发生器及同步合成电路组成，它是以主副画面交替显示的方式显示在画面的上端，因此，还设有一高速电子开关电路。具体电路原理框图如图 4-54 所示。这套装置既可以对照片进行伪彩色处理，也可以对底片进行处理。这些处理均是实时的。

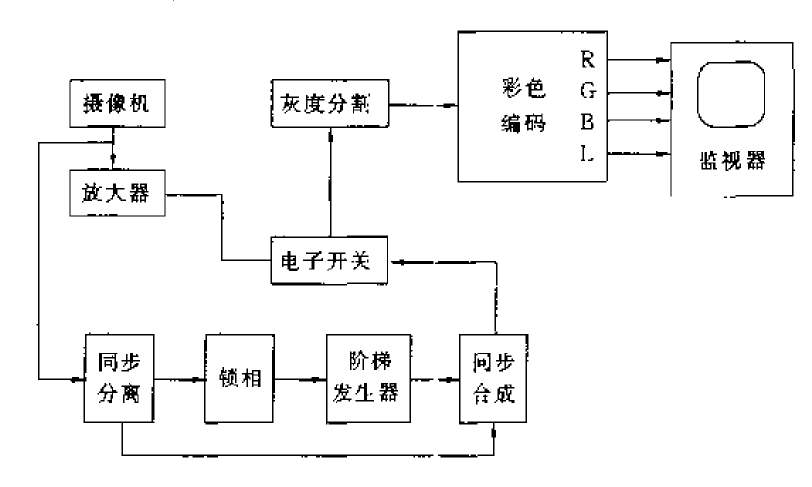


图 4-54 伪彩色增强装置原理框图

为满足分析不同图片的要求，在仪器设计上可采取如下措施：

1）在仪器输入端设计了一个成像工作台，在增强黑白照片时，用上光源照射照片，靠反射在摄像机上成像。当分析黑白底片时，靠下光源照射，经透射在摄像机上成像。

2）在分析黑白底片时，应注意透射率 T 和底片密度 D 之间的关系为

$$D = \lg \frac{1}{T}$$

这时如果不进行校正，则摄像机所成之图像的亮度与底片密度之间将不是线性关系，此时有些密度等级将会被压缩，有些密度等级会被拉长，可能会影响分析效果。因此，在分析底片时，应插入一个对数放大器进行校正，以便使底片密度 D 与输出信号呈线性关系。

3）照度总是有限的，因此，为提高分析精度，除了要求摄像机有较高的灵敏度外，还要求摄像管靶面尽可能的均匀。必要时也要加些补偿措施。

4）在分析图片时，往往会遇到想要判读的区域的色彩与周围色彩相近（如红和粉红），不

易分辨其边界，这时可利用改变彩色编码方式的方法来获得更易于辨识的色彩。

5）由于视频信号有大约6.5MHz的带宽，因此，为了提高图像的分辨率，放大器的带宽要足够。

6）彩色监视器的分辨率要高，会聚精度要好，几何失真要小，以便最大限度减小失真。

上述伪彩色增强仪不仅可独立做为处理设备使用，也可以与其他机器连用，如与计算机、医用X射线光电视、超声心动仪等设备连结作为一个显示终端。

伪彩色增强处理的计算机程序设计原理可见附录6。

程序利用VC5.0实现了灰度图像的16灰度级伪彩色显示。输入图像可以是任意大小、任意灰度级的，它被自动转换为16灰度级。灰度值和使用彩色的对应关系是：{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}→{黑、深蓝、深绿、深红、深灰、品红、浅蓝、棕色、浅绿、浅红、浅灰、浅蓝绿、黄色、白色、深蓝绿、浅紫}。其基本功能包括：

1）打开一灰度图像文件，显示该文件，并得到其宽度ImageWidth和高度ImageHeight；

2）读取图像数据，将灰度值存入二维数组Img[i][j]中。

3）求出最大灰度值maxGray，并将原图像灰度转换成16级灰度，即：

对任意一点(i,j)，计算其新的灰度值colorIndex=16.0*Img[i][j]/maxGray

4）根据新的灰度值查表确定其对应的颜色值。

5）显示彩色图像。

附录6　伪彩色增强处理程序实例

以下只列出了与具体处理有关的三个类中的消息响应函数。其中，OnFileOpen函数用于打开一个灰度图像文件，并得到其大小；OnDraw函数用于显示位图图像；OnColorize函数用于伪彩色处理。

```
//打开bmp灰度图像文件
void CEdgeDetectView::OnFileOpen()
{
    CString strTitle;
CFileDialog dlg(TRUE,"bmp","*.bmp");
if(dlg.DoModal()==IDOK){
    BmpFileName=dlg.GetFileName();
    BmpFileName.MakeUpper();
    if(m_pImageObject! =NULL)
        delete m_pImageObject;
    m_pImageObject=NULL;
    if(m_pImageObject==NULL){
        BeginWaitCursor();
        m_pImageObject=new CImageObject(BmpFileName);
        EndWaitCursor();
        if(m_pImageObject==NULL){
            AfxMessageBox("Could not create Image");
            return;
        }
```

```
        else {
            ImageWidth=m_pImageObject->GetWidth();
            ImageHeight=m_pImageObject->GetHeight();
        m_pImageObject->GetPaletteData(rgbQuad);
        strTitle="EdgeDetect-"+BmpFileName;
        AfxGetMainWnd()->SetWindowText(strTitle);
          CRect rect;
          GetClientRect(&rect);
          InvalidateRect(rect,TRUE);

    }
  }
}
/////////////////////////////////////////////////////////////////////////////
// CEdgeDetectView drawing
//显示位图
void CEdgeDetectView::OnDraw(CDC * pDC)
{
    CEdgeDetectDoc * pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    if(m_pImageObject! =NULL){
        if(GetFocus()==this)
            m_pImageObject->SetPalette(pDC);
        m_pImageObject->Draw(pDC);
    }
}
    //伪彩色处理
void CEdgeDetectView::OnColorize()
{
    // TODO: Add your command handler code here
    int i,j;
    COLORREF color;
//16 级彩色表,依次为:
      //black,dark blue,dark green,dark red,
      //dark gray,dark magenta,light blue,brown,
      //light green,light red,light gray,light cyan,
      //yellow,white,dark cyan,light magenta

      COLORREF pColor[16]=
      {
      0x00000000,0x00770000,0x00007700,0x00000077,
      0x003f3f3f,0x00770077,0x00ff0000,0x00007777,
      0x0000ff00,0x000000ff,0x00808080,0x00ffff00,
```

```
        0x0000ffff,0x00ffffff,0x00777700,0x00ff00ff
        };
        int colorIndex;
        int maxGray;
//*******读灰度图像BMP文件*********//
      CFile bmpfile;
      bmpfile.Open(BmpFileName,CFile::modeRead);
      if(bmpfile==NULL){
        AfxMessageBox("Could not Open the bmp file");
        return;
      }
      bmpfile.Seek(1078,CFile::begin);
      //read input file to Img[i][j]
      BYTE cTemp[640];
      for(j=0;j<ImageHeight;j++)
          {bmpfile.Read(cTemp,SIZE);
          //memcpy(Img[SIZE-1-i],cTemp,SIZE);
          for(i=0;i<ImageWidth;i++)Img[i][ImageHeight-1-j]=cTemp[i];
      }
      bmpfile.Close();

      CDC *dc=GetDC();
      //求最大像素值
      maxGray=0;
for(i=0;i<ImageWidth;i++)
for(j=0;j<ImageHeight;j++)
maxGray=maxGray<Img[i][j]?Img[i][j]:maxGray;

      // gray bmp image->16 color image
for(i=0;i<ImageWidth;i++)
for(j=0;j<ImageHeight;j++)
{
      //转换成16级灰度
      colorIndex=(int)(16.0*(double)Img[i][j]/(double)maxGray);
      //转换成相应的颜色
      if(colorIndex>=15)colorIndex=15;
      //查彩色表
      color=pColor[colorIndex];
          dc->SetPixel(SIZE+10+i,j,color); //显示彩色像素
}
          ReleaseDC(dc);
```

思 考 题

1. 图像增强的日的是什么?

2. 什么是直方图?

3. 直方图修改的技术基础是什么;

4. 在直方图修改技术中采用的变换函数的基本要求是什么?

5. 直方图均衡化处理采用何种变换函数?

6. 直方图均衡化处理的结果是什么?

7. 假定有 64×64 大小的图像,灰度为 16 级,概率分布如下表,试用直方图均衡化方法处理之,并画出处理前后的直方图。

r	n_k	$P_r(r_k)$
$r_0=0$	800	0.195
$r_1=\frac{1}{15}$	650	0.160
$r_2=\frac{2}{15}$	600	0.147
$r_3=\frac{3}{15}$	430	0.106
$r_4=\frac{4}{15}$	300	0.073
$r_5=\frac{5}{15}$	230	0.056
$r_6=\frac{6}{15}$	200	0.049
$r_7=\frac{7}{15}$	170	0.041
$r_8=\frac{8}{15}$	150	0.037
$r_9=\frac{9}{15}$	130	0.031
$r_{10}=\frac{10}{15}$	110	0.027
$r_{11}=\frac{11}{15}$	96	0.023
$r_{12}=\frac{12}{15}$	80	0.019
$r_{13}=\frac{13}{15}$	70	0.017
$r_{14}=\frac{14}{15}$	50	0.012
$r_{15}=0$	30	0.007

8. 如果有如下图所示的直方图的图像,它是否可用直方图均衡化方法处理?如果可以,其结果如何?如果不可以,采用什么方法可以解决该问题?

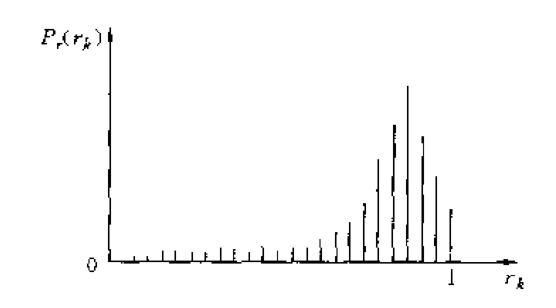
9. 直方图均衡化处理的主要步骤是什么?

10. 什么是"简并"现象?如何克服简并现象?

11. 直方图规定化处理的技术难点是什么?如何解决?

12. 试写一段直方图均衡化处理的程序。

13. 在邻域平均法处理中如何选取邻域?



14. 低通滤波法中通常有几种滤波器？它们的特点是什么？
15. 多图像平均法为何能去掉噪声，它的主要难点是什么？
16. 图像尖锐化处理有几种方法？
17. 零交叉边缘检测的优点是什么？
18. 何为同态处理？试述其基本原理。
19. 彩色图像的获取方法有几种？
20. 在图像处理中有哪几种彩色模型？它们的应用对象是什么？
21. 何为 RGB 模型？
22. 何为 CMY 模型？
23. 何为 YIQ 模型？
24. 何为 HIS 模型？
25. 如何由 RGB 模型转换为 HIS 模型？
26. 如何由 HSI 模型转换为 RGB 模型？
27. 如何由 RGB 模型转换为 YIQ 模型？
28. 如何由 YIQ 模型转换为 RGB 模型？
29. 如何由 RGB 模型转换为 CMY 模型？
30. 什么是伪彩色增强处理？它的主要目的是什么？
31. 试写一段伪彩色增强程序。

第 5 章　图像编码

模拟图像信号在传输过程中极易受到各种噪声的干扰，而且模拟图像信号一旦受到污染则很难完全得到恢复。另外，在模拟领域中，要进行人与机器(计算机或智能机)，机器与机器之间的信息交换以及对图像进行诸如压缩、增强、恢复、特征提取和识别等一系列处理都是比较困难的。所以无论从完成图像通信与数据通信网的结合方面来看，还是从对图像信号进行各种处理的角度来看，图像信号数字化都是首当其冲的重要问题。图像数字化的关键是编码。本章将对图像编码进行较为详细的讨论。

5.1　图像编码分类

编码是把模拟制信号转换为数字制信号的一种技术。图像编码技术不仅是应用线性脉冲编码调制法(线性 PCM)，而更主要的是利用图像信号的统计特性以及视觉对图像的生理学和心理学特性对图像进行信源编码。

在信息论中将通信过程概括为图 5-1 的形式。

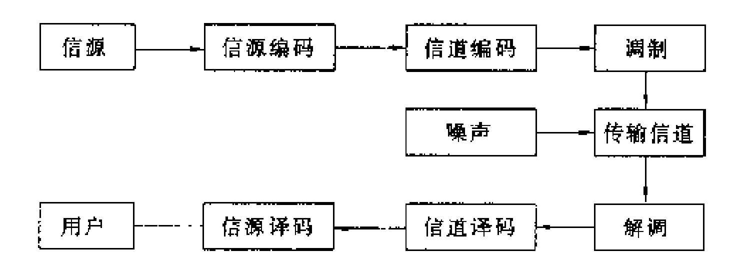


图 5-1　通信系统模型

图 5-1 所示的模型是一个数字通信系统模型。在这样一个模型中有二次编、译码，即信源编码、信源译码和信道编码、信道译码。信源编码的主要任务是解决有效性问题，也就是对信源实现压缩处理，使处理后的信号更适宜数字通信系统。解决有效性问题就是在编码过程中尽量提高编码效率，也就是力求用最少的数码传递最大的信息量。这实质上就是压缩频带的问题。信道编码的任务是解决可靠性问题。也就是尽量使处理过的信号在传输过程中不出错或少出错，既使出了错也要有能力尽量纠正错误。这是信道编码所应完成的任务。因此，在信道编码中往往引进用作误差控制的数码，以实现自动检错和纠错。因此，二次编、解码承担着不同的任务。图像编码主要是要研究压缩数码率，即高效编码问题。

编码是信息处理科学中的经典研究课题，就图像编码而言，已有近 50 年的历史，近年来，Kunt 提出第一代、第二代编码的概念。Kunt 把 1948—1988 年 40 年中研究的以去除冗余为基础的编码方法称为第一代编码，如：PCM、DPCM、ΔM、亚取样编码法，变换域的 DFT、DCT、Walsh-Hadamard 变换编码等方法以及以此为基础的混合编码法均属于经典的第一代编码法。而第二代编码方法多是 80 年代以后提出的新的编码方法，如金字塔编码法、Fractal 编码、基于神经元网络的编码方法、小波变换编码法、模型基编码法等。现代编码法的特点是：①充分

考虑人的视觉特性；②恰当地考虑对图像信号的分解与表述；③采用图像的合成与识别方案压缩数据率。这种分法尽管并没得到图像编码界全体同仁的广泛认可，但笔者认为对了解图像编码发展进程是有益的。

图像编码这一经典的研究课题，经 50 余年的研究已有多种成熟的方法得到应用，特别是所谓的第--代编码更是如此。随着多媒体技术的发展，已有若干编码标准由 ITU-T 制定出来，如 JPEG、H.261、H.263、MPEG1、MPEG2、MPEG4、MPEG7、JBIG (Joint Bi-level Image Coding Expert Group，二值图像压缩)等。相信经广大科技工作者的不懈努力，在未来会有更多、更有效的编码方法问世，以满足多媒体信息处理及通信的需要。

图像编码属于信源编码的范畴。对它的方法进行归类并不统一，从不同的角度来看问题就会有不同的分类方法。按图像形式可分为图示像和非图示像的编码。从光度特征出发可分为单色图像、彩色图像和多光谱图像编码。从灰度层次上分可以分为二值图像和多灰度图像编码，按照信号处理形式可分为模拟图像编码和数字图像编码。从处理维数出发可以分成行内编码、帧内编码和帧间编码。从上面的介绍可见，图像编码的分类并没有统一的标准，同时也可以看到图像编码的方案也是多种多样的。图像编码是属于信源编码的范畴，从信源编码的角度来分类，图像编码大致可分为匹配编码、变换编码和识别编码。从压缩的角度来分，也可大致分为熵压缩编码和无失真编码。

如果从目前已有的实用方案的角度来分类，可以分为三大类：即预测编码，变换编码及统计编码。而这些方法既适用于静止图像编码，也适用于电视信号编码。就具体编码方法而言可简略地概括在图 5-2 中。

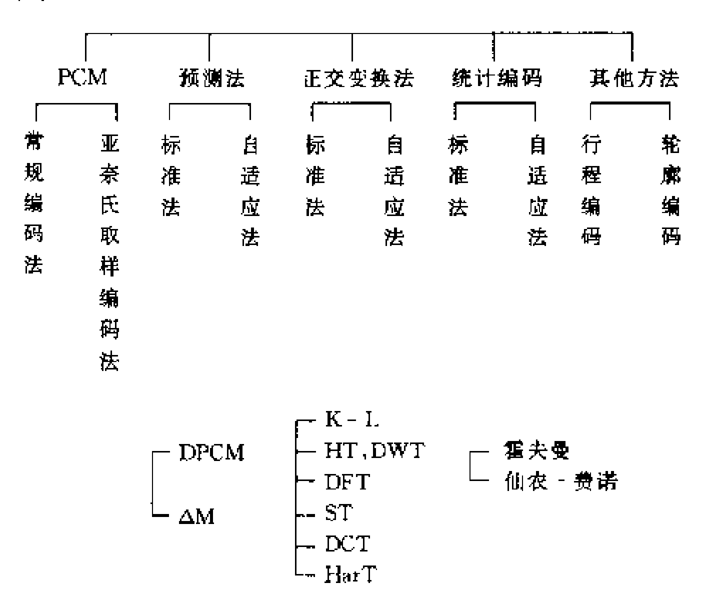


图 5-2　图像高效编码法

最后需要着重提及的是，上述各种具体方案并不是孤立的、单一的使用，往往是各种方法重迭、交叉使用，以达到更高的编码效率，在 ITU-T 的建议标准中这一点尤为突出。

5.2　图像编码中的保真度准则

图像信号在编码和传输过程中会产生误差，尤其是在熵压缩编码中，产生的误差应在允许的范围之内。在这种情况下，保真度准则可以用来衡量编码方法或系统质量的优劣。通常，这

种衡量的尺度可分为客观保真度准则和主观保真度准则。

5.2.1 客观保真度准则

通常使用的客观保真度准则有输入图像和输出图像的均方根误差；输入图像和输出图像的均方根信噪比两种。输入图像和输出图像的均方根误差是这样定义的，设输入图像是由 $N\times N$ 个像素组成，令其为 $f(x,y)$，其中 $x,y=0,1,2,\cdots,N-1$。这样一幅图像经过压缩编码处理后，送至受信端，再经译码处理，重建原来图像。这里令重建图像为 $g(x,y)$。它同样包含 $N\times N$ 个像素，并且 $x,y=0,1,2,\cdots,N-1$。在 $0,1,2,\cdots,N-1$ 范围内 x,y 的任意值，输入像素和对应的输出图像之间的误差可用下式表示：

$$e(x,y)=g(x,y)-f(x,y) \tag{5-1}$$

而包含 $N\times N$ 像素的图像之均方误差为

$$\begin{aligned}\bar{e}^2&=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}e^2(x,y)\\&=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[g(x,y)-f(x,y)]^2\end{aligned} \tag{5-2}$$

由式(5-2)可得到均方根误差为

$$e_{\mathrm{rms}}=[\bar{e}^2]^{\frac{1}{2}} \tag{5-3}$$

如果把输入、输出图像间的误差看作是噪声，那么，重建图像 $g(x,y)$ 可由下式表示：

$$g(x,y)=f(x,y)+e(x,y) \tag{5-4}$$

在这种情况下，另一个客观保真度准则——重建图像的均方信噪比如下式表示：

$$\begin{aligned}\left(\frac{S}{N}\right)_{\mathrm{ms}}&=\frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}g^2(x,y)}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}e^2(x,y)}\\&=\frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}g^2(x,y)}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[g(x,y)-f(x,y)]^2}\end{aligned} \tag{5-5}$$

均方根信噪比为

$$\left(\frac{S}{N}\right)_{\mathrm{rms}}=\left\{\frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}g^2(x,y)}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[g(x,y)-f(x,y)]^2}\right\}^{\frac{1}{2}} \tag{5-6}$$

5.2.2 主观保真度准则

图像处理的结果，绝大多数场合是给人观看，由研究人员来解释的，因此，图像质量的好坏与否，既与图像本身的客观质量有关，也与人的视觉系统的特性有关，有时候，客观保真度完全一样的两幅图像可能会有完全不相同的视觉质量，所以又规定了主观保真度准则。这种方法是把图像显示给观察者，然后把评价结果加以平均，以此来评价一幅图像的主观质量。另外一种方法是规定一种绝对尺度，例如：

1）优秀的　具有极高质量的图像；

2）好的　是可供观赏的高质量的图像，干扰并不令人讨厌；

3）可通过的　图像质量可以接受，干扰不讨厌；

4）边缘的　图像质量较低，希望能加以改善，干扰有些讨厌；

5）劣等的　图像质量很差，尚能观看，干扰显著地令人讨厌；

6）不能用　图像质量非常之差，无法观看。

另外常用的还有两种准则，即妨害准则和品质准则。

妨害准则如下五级：

1）没有妨害感觉；

2）有妨害，但不讨厌；

3）能感到妨害，但没有干扰；

4）妨害严重，并有明显干扰；

5）不能接收信息。

品质准则如下七级：

1）非常好；

2）好；

3）稍好；

4）普通；

5）稍坏；

6）恶劣；

7）非常恶劣。

除此之外，还可以采用成对比较法，也就是同时出示两幅图像，让观察者表示更喜欢哪一幅图像，借此排出图像质量的等级。也有采用随机抽取法来评定图像质量的。这种方法是把数量相等的原始图像和经编译码后的图像混杂在一起，然后让观察者挑出他认为质量差的图像。质量较差的图像可定义为处理过的图像，然后统计错挑的概率，显然错挑概率越大说明图像经处理后的劣化越小。总之，主观保真度评价方法的准则可不同，但其基本原理都一样，当然，对观察者的视觉条件应有一定的要求。

5.3 PCM 编码

5.3.1 PCM 编码的基本原理

脉冲编码调制（Pulse Coding Modulation，PCM ）是将模拟图像信号变为数字信号的基本手段。图像信号的 PCM 编码与语音信号 PCM 编码相比并没有原则上的区别。但是，图像信号，特别是电视信号占的频带宽，要求响应速度快，因此，电路设计与实现上有较大的难度。

图像信号 PCM 编、译码原理方框图如图 5-3 所示。图像信号 PCM 编码由前置低通滤波器、取样保持电路、量化器、编码器组成。前置低通滤波器的作用有两个，一是为满足取样定理的带限要求，以减少折迭误差；其二是对杂散噪声也有一定的抑制作用。取样保持电路将完成把时间上是连续的模拟信号进行时间离散化的任务。取样周期由奈奎斯特（Nyquist）定理限定。

量化器的任务是把模拟信号的幅值离散化。经取样与量化处理后就可产生多值数字信号。

编码的任务是把多值的数字信号变成二进制数字的多比特信号，以便传输或进行后续处理。译码器的原理比较简单，它包括一个译码电路和一个低通滤波器。译码器把数字信号恢复为模拟信号，这个模拟信号就是在接收端重建的图像信号。

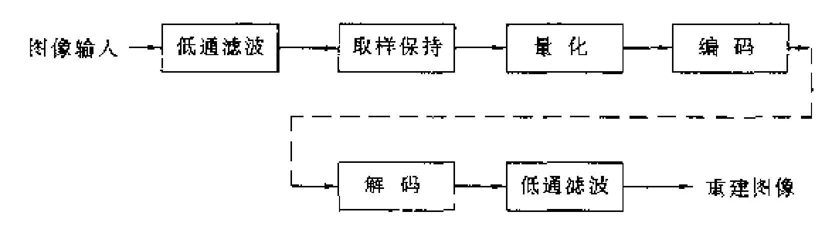


图 5-3 PCM 编、译码原理方框图

5.3.2 PCM 编码的量化噪声

量化是对时间离散的模拟信号进行幅度离散化的过程，这个过程是去零取整的过程，量化后的样值与原信号相比大部分是近似关系。这样，把连续的数值限制在固定的台阶式的变化之下必然会带来畸变。这种畸变在接收端是无法克服的，只能使其尽量减小。由量化带来的噪声可分为量化噪声和过载噪声。以正弦波输入为例，输入幅度较大和输入幅度较小时的量化噪声如图 5-4 所示。图中(a)是输入信号超过编码范围时的量化噪声和过载噪声的形成，(b) 是信号未超过编码范围，只有量化噪声的情况。在 PCM 编码中，量化噪声主要取决于码的位数，码位数越多(即量化阶数多)量化噪声的功率越小。一个量化阶的电压可由下式表示：

$$\Delta = \frac{V}{2^n} \tag{5-7}$$

式中，V 为输入信号电压；n 为样值用二进制数表示的比特数。

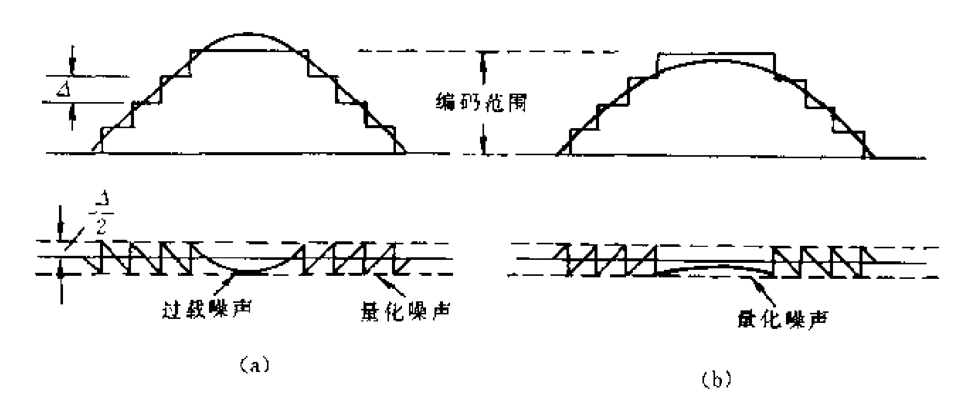


图 5-1 量化噪声与过载噪声的形成

如果在整个输入幅度内量化阶是一个常数，就称这个量化为均匀量化，否则就是非均匀量化。线性 PCM 编码中均采用均匀量化法。在均匀量化中，设量化阶为 Δ，量化噪声在 $-\frac{\Delta}{2}$～$+\frac{\Delta}{2}$ 内可看成是均匀分布的，因此，其功率可由下式表示：

$$P_Q = \int_{-\frac{\Delta}{2}}^{+\frac{\Delta}{2}} \frac{1}{\Delta} x^2 \mathrm{d}x = \frac{\Delta^2}{12} \tag{5-8}$$

对于过载噪声，当量化特性输入过载点为 V 时，由下式表示：

$$N_S = \int_{-\infty}^{-V} (x+V)^2 p(x)\mathrm{d}x + \int_{V}^{+\infty} (x-V)^2 p(x)\mathrm{d}x \tag{5-9}$$

式中N_c为过载噪声，x是输入信号值，$p(x)$为输入幅度的概率密度。如果用信噪比作为客观保真度准则的话，可推得 PCM 编码在均匀量化下的量化信噪比如下：

因为

$$\Delta = \frac{V}{2^n}$$

$$P_Q = \frac{\Delta^2}{12}$$

所以

$$P_Q = \frac{\frac{V^2}{(2^n)^2}}{12} \tag{5-10}$$

由信噪比的概念，则

$$\begin{aligned}\left(\frac{S}{N}\right)_{dB} &= 10\lg\frac{V^2}{P_Q}\\ &= 10\lg\frac{V^2}{\frac{\frac{V^2}{(2^n)^2}}{12}}\\ &= 10\lg[12\times(2^n)^2]\\ &= 20\lg\sqrt{2}+20\lg 2^n\\ &\approx (11+6n)\end{aligned} \tag{5-11}$$

由式(5-11)可见，每增加一位码可得到 6dB 的信噪比得益。

值得注意的是，量化噪声不同于其他噪声，它的显著特点是仅在有信号输入时才出现，所以它是数字化中特有的噪声。一般情况下，直接测量比较困难。

5.3.3 编码器、译码器

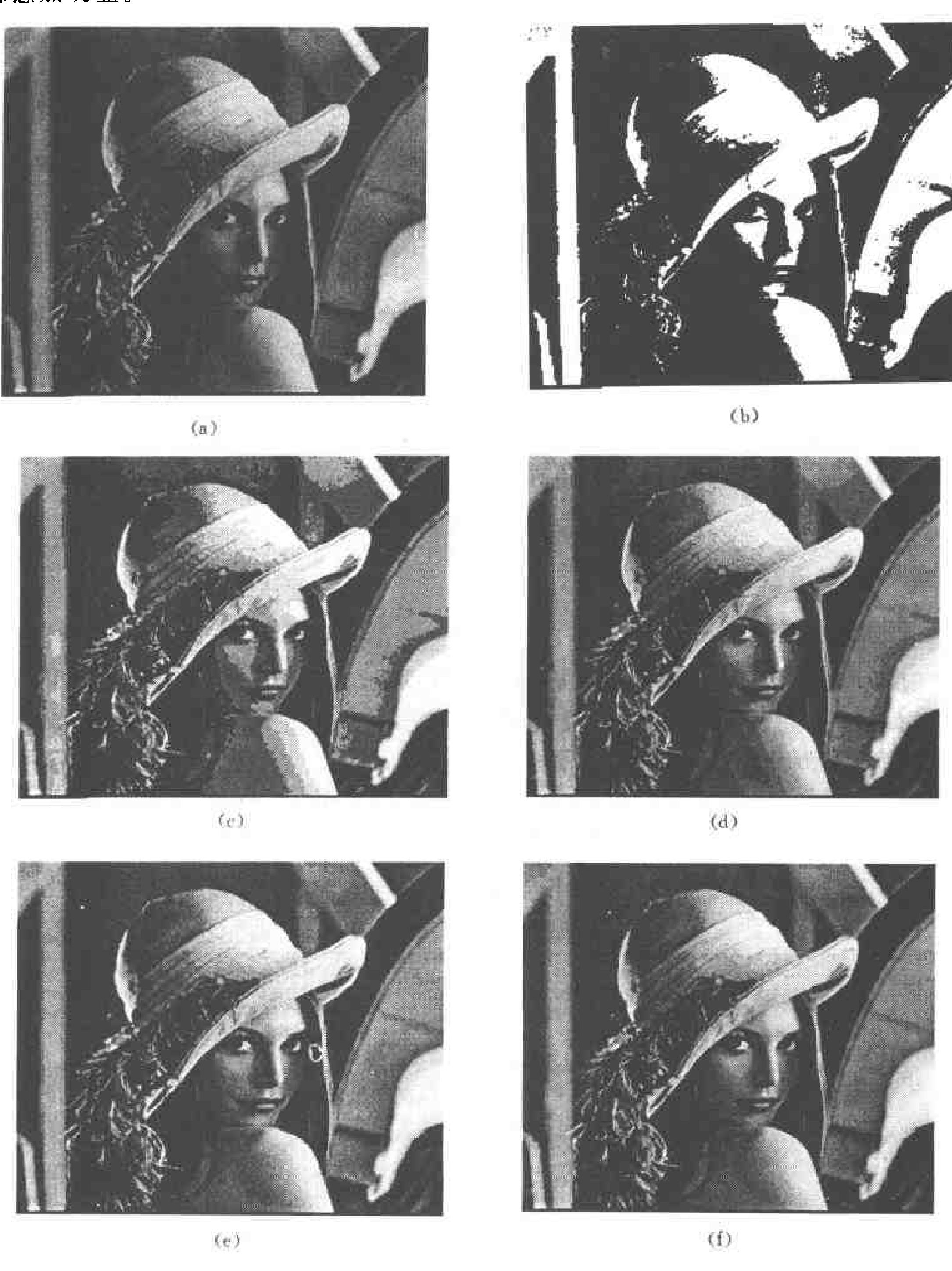
编码器的任务是把一个多值的数字量用多比特的二进制量来表示。如果量化器输出 M 个值，那么，对应于 M 个值中的任何一个值编码器将给定一个二进制码字。这个码字将由 m 个二进制数组成。通常情况下 $M=2^m$。编码器的输入与输出关系是一一对应的，其过程是可逆的，因此，不会引入任何误差。

线性 PCM 编码一般采用等长码，也就是说每一个码字都有相同的比特数。其中用得最为普遍的是自然二进码，也有用格雷码的。以 $M=8$ 为例的自然二进码和格雷码列入表 5-1。

表 5-1 $M=8$ 的自然二进码和格雷码

输入	自然二进码	格雷码
m_1	0 0 0	0 0 0
m_2	0 0 1	0 0 1
m_3	0 1 0	0 1 1
m_4	0 1 1	0 1 0
m_5	1 0 0	1 1 0
m_6	1 0 1	1 1 1
m_7	1 1 0	1 0 1
m_8	1 1 1	1 0 0

对于常规编码来说，减少量化分层数就可以降低比特速率。但是，量化分层数的最小值应满足图像质量的要求。当主观评定图像质量时，为了防止伪轮廓效应，量化分层数必须足够大。实践证明，对于线性 PCM 编码，黑白图像要 6～7bit，相当于分层数为 64～128 层，彩色复合编码要 8bit，即 256 个量化分层，这样恢复的图像才能与原模拟图像相比拟。如果码位不够，就会出现明显的伪轮廓。图 5-5 分别是女孩头像的编译码结果。图中(a) 是原像，(b)是 1bit 编译码处理后的图像，(c)是 2bit 编译码处理后的图像，(d)是 3bit 编译码处理后的图像，(e)是 4bit 编译码处理后的图像，(f)是 6bit 编译码处理后的图像，从图像上可见，随着比特数的降低伪轮廓愈加明显。



(a) (b) (c) (d) (e) (f)

图 5-5 编码位数对画面质量的影响

译码器通常采用电流相加式译码网，原理图如图 5-6 所示。在这种方案中，每一位用一个独立的电流源，可以通过分别调整每一位的恒流源来补偿网络误差，因此，这种电路对电阻网的精度要求可适当放宽。这种方案的另一个优点是开关的接触电阻不会影响节点上的电压，因此，对开关电路的要求不高。电流相加式译码网由三部分组成，这三部分是 T 型电阻网、电子开关、恒流源。总输出电压由下式表示：

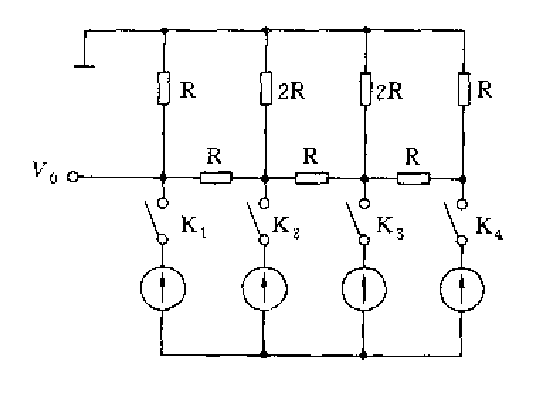


图 5-6 电流相加式译码原理

$$V_o = \sum_{i=0}^{n} a_i \cdot 2^i \cdot E_0 \qquad (5\text{-}12)$$

式中 a_i 等于 1 或 0，当第 i 位开关闭合时则为 1，否则为 0。E_0 是开关闭合时的输出电压，

$$E_0 = \frac{2}{3}RI\left(\frac{1}{2}\right)^n \qquad (5\text{-}13)$$

I 是恒流源输出电流。译码电路的电原理图如图 5-7 所示。

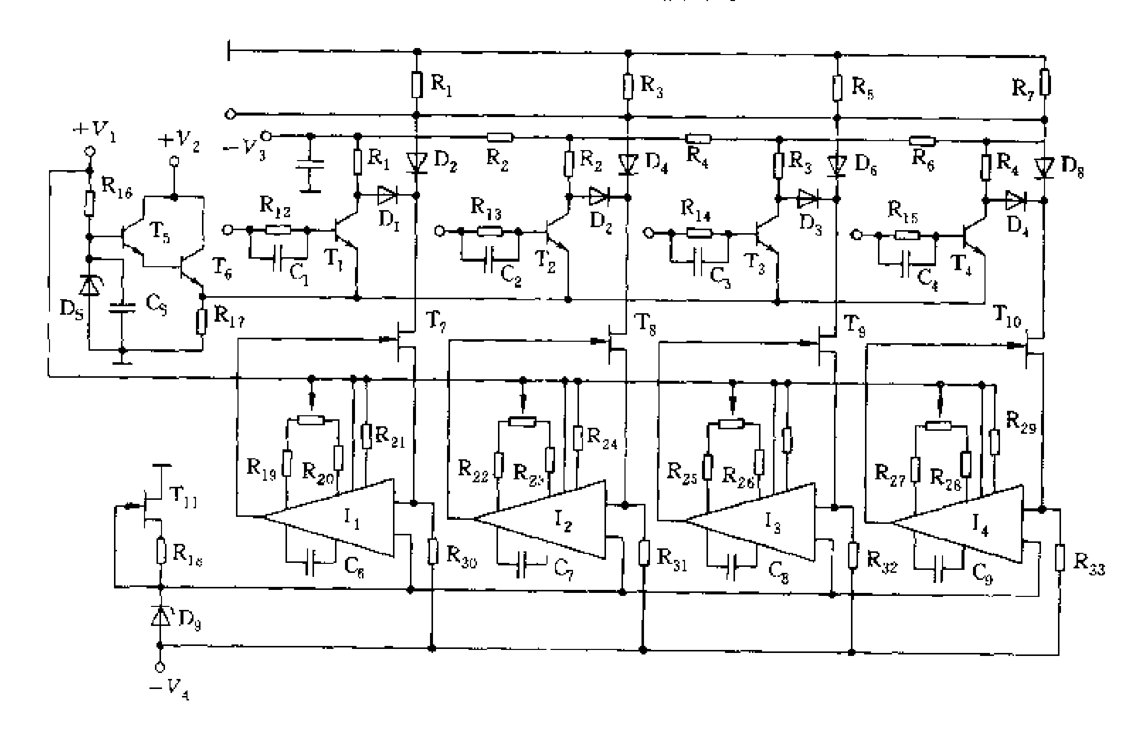


图 5-7 译码电路电原理图

5.3.4 非线性 PCM 编码

在线性 PCM 编码中，量化阶是均匀的。这样，在小信号输入的情况下信噪比较低，在大信号输入的情况下信噪比较高。为了改善小信号在量化过程中的信噪比，采用一种瞬时压缩扩张技术。这种技术实际上是降低大信号时的信噪比提高小信号时的信噪比，其结果是在不增加数码率的情况下，使信号在整个动态范围内有较均衡的信噪比。

瞬时压扩技术基本有两种方案。一种是如图 5-8 所示的方案。在这种方案中，首先将取样后的 PAM 信号进行非线性压缩，然后对压缩后的 PAM 信号进行线性编码。在接收端，首先进

行线性译码，然后再送入瞬时扩张器进行非线性扩张，恢复原来的 PAM 信号。

该方案一般采用二极管电压、电流的非线性特性来实现。通常采用图 5-9 所示的修正的对数特性。在图 5-9 中，对于压缩器来说，X 表示输入，Y 表示输出。对于扩张器来说，Y 表示输入，X 表示输出。参变量 μ 是表示压缩或扩张程度的量。显然，当 $\mu=0$ 时就是线性编码。

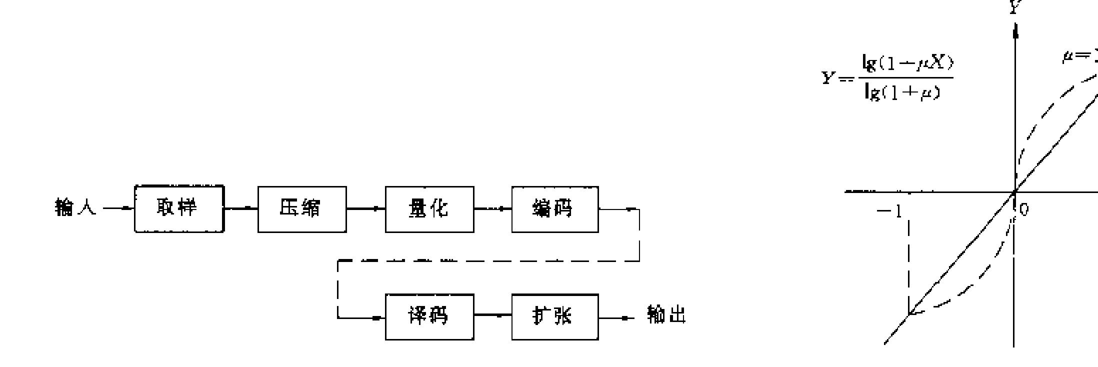


图 5-8　一种非线性压扩方案　　　　图 5-9　理论压扩曲线

这种方案电路比较简单，但是，要保证压缩与扩张特性的匹配比较困难。特别是在温度变化范围很宽的情况下必须有恒温设备，因此，给生产和调试均增加很大麻烦。

另外一种方案是数字化非线性压扩技术，其原理框图如图 5-10 所示。这种方案是把编码与压缩，译码与扩张都分别在编码和译码中一次完成。数字式非线性编码的压扩特性有 μ 特性、A 特性等等。根据 CCITT 1970 年的建议，通常采用 13 折线（$A=87.6$）的压扩特性。13 折线压扩特性如图 5-11 所示。各折线的斜率列于表 5-2 中。由图中可见，各段折线的斜率是不一样的；4 至 8 段的小信号区的信噪比都得到了改善。图中的 u_{in} 表示压缩器的输入，u_{out} 表示压缩器的输出，V 为过载点电压。图 5-11 只画出了信号在正半周时的情况，负半周时也一样。由于正半周的 7、8 两段和负半周的 7、8 两段斜率都一样，所以在整个特性中这 4 段连成一条直线。因此，总共有 13 条直线段，简称 13 折线。

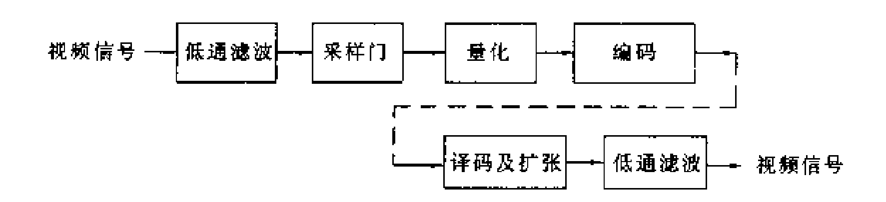


图 5-10　数字化非线性压扩技术框图

表 5-2　各种线段斜率表

折线段	1	2	3	4	5	6	7	8
斜率	$\frac{1}{1}$	$\frac{1}{2}$	1	2	4	8	16	16

如果令 $x=\frac{u_{in}}{V}$，$y=\frac{u_{out}}{V}$，上述 13 折线可用下式近似表示：

$$y = \frac{Ax}{1 + \ln A} \qquad 0 < x \leqslant \frac{1}{A} \tag{5-14}$$

$$y = \frac{1 + \ln Ax}{1 + \ln A} \qquad \frac{1}{A} < x \leqslant 1 \tag{5-15}$$

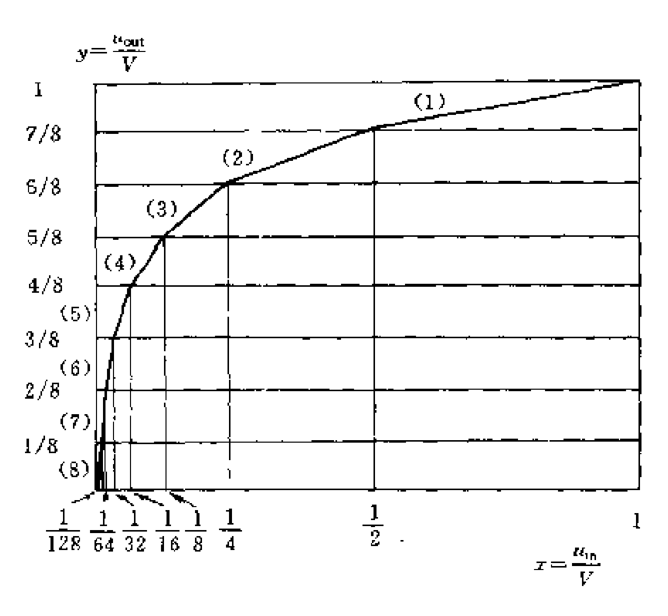


图 5-11　13 折线压扩特性(信号为正时的 8 段)

式中 A 是一常数,不同的 A 值可决定一条不同的曲线。在原点处的斜率由下式表示:

$$\frac{dy}{dx} = \frac{A}{1 + \ln A} \tag{5-16}$$

当原点处的斜率为 16 时,则

$$\frac{dy}{dx} = \frac{A}{1 + \ln A} = 16$$

可求得 $A=87.6$。图 5-11 所示的折线就是 $A=87.6$ 的 13 折线压缩特性。

总的来说,如果采用第一种方案,当 $\mu=100$ 时,在输入信号最小情况下信噪比可提高 26dB;当采用第二种方案时,在 $A=87.6$ 的情况,最小信号输入时的信噪比改善约 24dB。可见,两种方案的效果基本相似,在小信号输入情况下,信噪比的改善是很显著的。数字化非线性压扩技术的电路实现比第一方案要复杂。但随着数字集成电路的发展,电路制做方面的障碍已不复存在。它能较容易地实现压扩特性的匹配,而且比较稳定可靠,不用恒温设备,在生产中也可以保证编、译码的精度,调试与维护也较简单,因此,这种方案用得愈来愈广泛。

5.3.5　亚奈奎斯特取样 PCM 编码

线性 PCM 编码是最基本的数字化手段,数字化处理时取样速率必需满足奈奎斯特取样定理的要求,否则,会产生混叠误差而不能在收端恢复原图像信息。取样定理可由下式表示:

$$f_s \geqslant f \times 2 \tag{5-17}$$

式中 f 是模拟信号的频带宽度,f_s 是取样速率。PCM 编码的速率由下式表示:

$$R = 2f_s n \tag{5-18}$$

式中 n 是每样值的比特数。从式(5-18)可见,降低 R 的方法有二个,一是降低 n 值,其次是降低取样速率。非线性 PCM 方法实质上是用非线性措施尽量减小 n 值。也就是说在尽可能小的 n

值下尽量改善画面的质量。鉴于对画面灰度层次的要求，非线性 PCM 编码对 R 的减少是有限的。

能否用降低取样率的方法来减少 R 呢？如果能解决混叠误差问题，那么，这种设想也是可行的。经分析，电视信号的频谱如图 5-12 所示。由图可见，电视图像信号的能量不是均匀连续地分布在整个带宽范围内，而是集中在以行频的各次谐波为中心的一束束谱线内。其最高谐波次数为 $352f_h$（以 5.5MHz 带宽计）。这里 f_h 代表行频。如果用低于奈奎斯特取样速率的频率取样，必然会产生折叠噪声。如果能够使折叠部分如图 5-13 所示那样落在原信号频谱的间隙内，在接收端再采用梳状滤波器将这些折叠噪声滤除，也可以消除折叠噪声，正确恢复原来图像。这就是亚奈奎斯特取样 PCM 编码的基本原理。亚奈奎斯特取样 PCM 编码的原理方框图如图 5-14 所示。

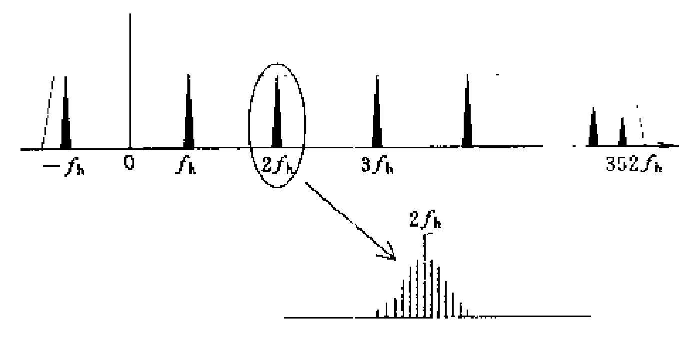


图 5-12　电视信号的谱

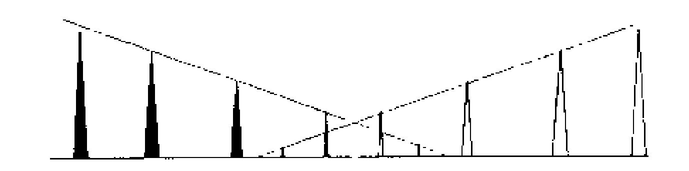


图 5-13　亚奈奎斯特取样频谱折叠部分的设置

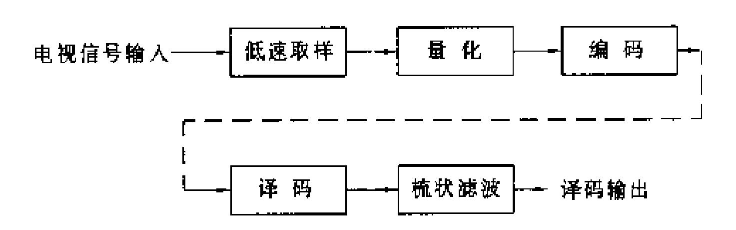


图 5-14　亚奈奎斯特取样 PCM 编码原理方框图

这种方案在单色电视信号或彩色电视信号各分量信号编码中，可选择取样频率等于半行频的奇数倍，即

$$f_s = (2m + 1) \cdot \frac{f_h}{2} \tag{5-19}$$

式中 f_s 代表取样频率，f_h 代表行频，m 取正整数。这样，就可以使频谱折叠部分落入原信号频谱的间隙内。采用这种方案，大约可节省 30% 的数码率。

如果直接对复合彩色电视信号编码，由于在频谱中含有接近单纯正弦波的彩色副载波，它的量化噪声与取样频率关系密切，如果落入图像频带内会产生辉度和色度的拍频干扰。为了使拍频落入带外，应将取样频率选为 3 倍或 4 倍的彩色副载波频率，在这种情况下，称作常规 PCM 编码的设计。如果选 2 倍的副载波频率为取样频率，则也属于低速取样范畴。罗塞尔曾对

NTSC 制彩色电视信号进行亚奈奎斯特取样 PCM 编码实验。这套系统是在 CRS 技术中心研制的。NTSC 制彩色电视信号的频谱是这样分布的，亮度信号的频谱能量基本集中在行频 f_h 上，色度信号的频谱能量集中在半行频的奇次谐波上，即在 $\left(n+\frac{1}{2}\right)f_h$ 上，而且色度和亮度谱束交错相间。为了采用亚奈奎斯特取样 PCM 编码法，采取如下取样速率：

$$f_s = 2f_{sc} + \frac{f_h}{4} \tag{5-20}$$

或者

$$f_s = 2f_{sc} - \frac{f_h}{4} \tag{5-21}$$

式中 f_{sc} 是 NTSC 彩色副载波，f_h 是行扫描频率，f_s 是取样频率。在译码输出端所使用的梳状滤波器，频率在 f_s-f_v 和 f_v 之间，这样可滤除混叠噪声。f_v 是 NTSC 制视频信号带宽。这种系统的梳状滤波器设计是很关键的，在这个系统中采用的是横向梳状滤波器。当取样频率为 $2f_{sc}+\frac{f_h}{4}$ 时，基带亮度信号 Y_B 和色度信号 C_B 以及混叠分量 Y_A 和 C_A 的主要谱线如图 5-15 所示。

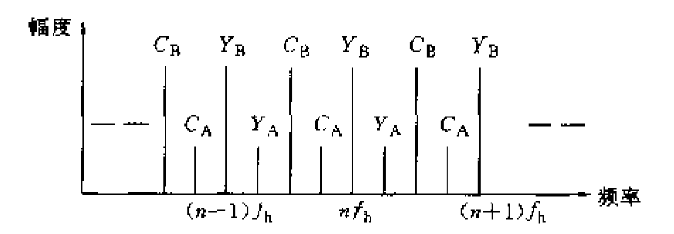


图 5-15 在亚奈奎斯特编码速率为 $2f_{sc}+\frac{f_h}{4}$ 时亮度信号 Y_B 和色度信号 C_B 以及混叠分量 Y_A、C_A 的谱图

示。显然，混叠分量 C_A、Y_A 是从 $\frac{f_h}{2}$ 的频率间隔分开的，因此，梳状滤波器在 $\frac{f_h}{2}$ 频率间隔中有最大响应或最小响应。把视频信号与另一时间序列 TV 行合并起来，就可构成梳状滤波器。在对 NTSC 彩色电视信号的亚奈奎斯特编码中可采用如下滤波算法：将电视信号第 l 行加到第 $(l-2)$ 行；将电视信号第 1 行加到第 $(l+2)$ 行（这两种算法的频率响应如图 5-16 所示）；将电视信号第 l 行加到第 $\frac{1}{2}[(l-2)+(l+2)]$ 行。这三种算法的所有行都取自一场。利用两行 TV 的梳状滤波器原理框图如图 5-17，图(a)为低频不延迟，(b)为低频延迟两行扫描线。

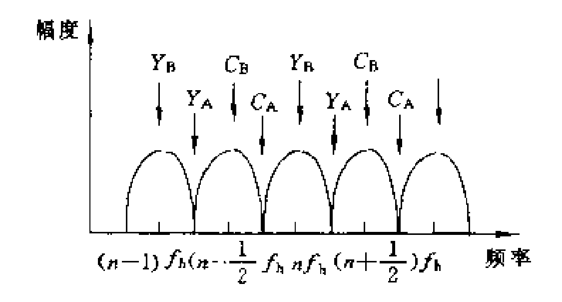


图 5-16 采用 $l+(l-2)$ 或 $l+(l+2)$ 算法的梳状滤波器频响

另外，对彩色电视信号进行 PCM 编码时，可采用分量编码法。因为对 R,G,B 三个信号进

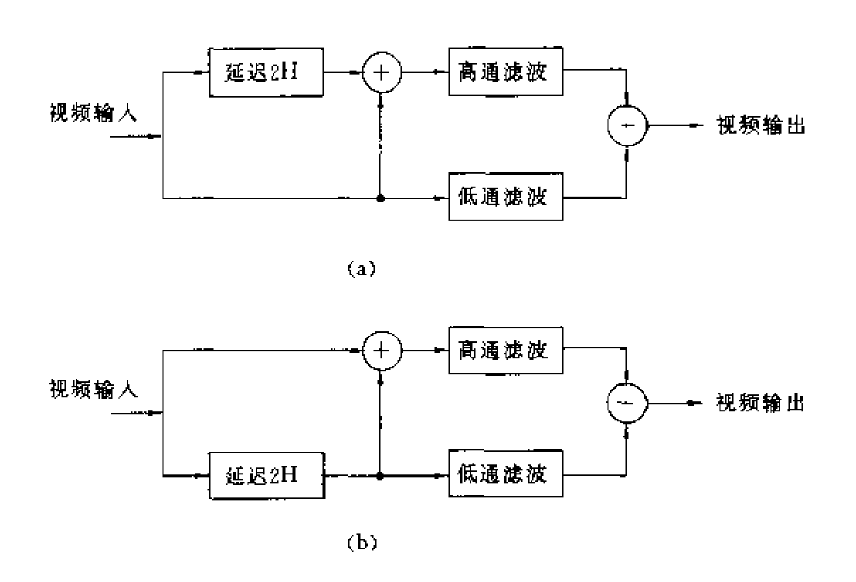


图 5-17 利用两行 TV 的梳状滤器原理框图

行数字化时，并不需要用相同的精度去量化它们。这三个分量对外加噪声敏感程度并不一样。例如，显示亮度恒定的情况下，它们的噪声门限如下：蓝色图像为 36dB，红色图像为 41dB，绿色图像为 43dB。对正常图像来讲，三个彩色分量的噪声灵敏度的差别就更大。在蓝色信号中，噪声能见度比红色小 10dB，比绿色小 20dB。为了对全带宽信号量化而不产生明显误差，蓝色信号需 4bit，红色信号需 5bit，绿色信号需 6bit。显然，考虑到这些因素，在分量编码中也有减少数码率的潜力。

5.4 统计编码

高效编码的主要方法是尽可能去除信源中的冗余成份，从而以最少的数码率传递最大的信息量。冗余度存在于像素间的相关性及像素值出现概率的不均等性之中。对于有记忆性信源来说首先要去除像素间的相关性，从而达到压缩数码率的目的。对于无记忆性信源来说，像素间没有相关性，可以利用像素灰度值出现概率的不均等性，采用某种编码方法，也可以达到压缩数码率的目的。这种根据像素灰度值出现概率的分布特性而进行的压缩编码叫统计编码。

5.4.1 编码效率与冗余度

为了确定一个衡量编码方法优劣的准则，首先讨论一下编码效率与冗余度的问题。设某个无记忆信源共有 M 个消息，记作$\{u_1,u_2,\cdots,u_M\}$，其中消息 $u_i(i=1,2,\cdots,M)$各自出现的概率分别为$\{P_1,P_2,\cdots,P_M\}$。可把这个信源用下式表示：

$$X=\begin{Bmatrix}u_1,u_2,\cdots,u_M\\P_1,P_2,\cdots,P_M\end{Bmatrix} \tag{5-22}$$

根据该信源的消息集合，在字母集 $A=\{a_1,a_2,\cdots,a_n\}$中选取 a_i 进行编码。一般情况下取二元字母集 $A\in\{1,0\}$。通常，这一离散信源中的各个消息出现的概率并不相等。根据信息论中熵的定义，可计算出该信源的熵如下式：

$$H(X)=-\sum_{i=1}^{M}P_i\log_a P_i \tag{5-23}$$

式中 $H(X)$代表熵，P_i 代表第 i 个消息出现的概率。

例如，设一离散信源如下：

$$X=\begin{Bmatrix} u_1 & u_2 & u_3 & u_4 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{Bmatrix}$$

由式(5-23)可算出该信源的熵：

$$\begin{aligned} H(X) &= -\sum_{i=1}^{4} P_i \log_2 P_i \\ &= -\frac{1}{2}\log_2\frac{1}{2}-\frac{1}{4}\log_2\frac{1}{4}-\frac{1}{8}\log_2\frac{1}{8}-\frac{1}{8}\log_2\frac{1}{8}=\frac{7}{4}\text{ 比特/消息} \end{aligned}$$

设对应于每个消息的码字由 N_i 个符号组成。也就是说每个消息所对应的码字长度各为 N_i。那么，每个消息的平均码长可用下式表示：

$$\overline{N}=\sum_{i=1}^{M} P_i N_i \tag{5-24}$$

式中 $\overline{N}$ 代表平均码长，M 为信源中包含的消息的个数，P_i 为第 i 个消息出现的概率，N_i 为第 i 个消息对应的码长。平均而言，每个符号所含有的熵为

$$S=\frac{H(X)}{\overline{N}} \tag{5-25}$$

编码符号是在字母集 A 中选取的。如果编码后形成一个新的等概率的无记忆信源，字母数为 n，那么，它的最大熵应为 $\log_a n$ 比特/符号，因此，这是极限值。如果 $\frac{H(X)}{\overline{N}}=\log_a n$，则可认为编码效率已达到 100%，若 $\frac{H(X)}{\overline{N}}<\log_a n$，则可认为编码效率较低。

由上述概念，编码效率如下式表示：

$$\eta=\frac{H(X)}{\overline{N}\log_a n} \tag{5-26}$$

式中 η 代表编码效率，$H(X)$ 为信源的熵，$\overline{N}$ 为平均码长，n 为字母集合中的字母数。

如果以比特(bit)作单位，log 的底为 2，根据上述定义，则

$$\frac{H(X)}{\overline{N}}=\log_2 n \qquad \eta=100\%$$

$$\frac{H(X)}{\overline{N}}<\log_2 n \qquad \eta<100\%$$

显然，如果 $\eta\neq100\%$，就说明还有冗余度。因此，冗余度如下式表示：

$$R_d=1-\eta=\frac{\overline{N}\log_2 n-H(X)}{\overline{N}\log_2 n} \tag{5-27}$$

统计编码要研究的问题就在于设法减小 $\overline{N}$，使 η 尽量趋近于 1，R_d 趋近于 0。显然 $\overline{N}$ 值有一个理论最低限，当 $\eta=1$ 时，$\overline{N}$ 的最低限就是 $H(X)/\log_2 n$。可以根据这一准则来衡量编码方法的优劣。下面举例加以说明。

例：一个信源 X 和一个字母集合 A 如下：

$$X=\begin{Bmatrix} u_1 & u_2 & u_3 & u_4 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{Bmatrix}$$

$$A=\{0,1,2,3\}$$

可求得信源 X 的熵为

$$H(X) = \frac{7}{4} \text{ 比特/消息}$$

平均码长为

$$\overline{N} = 1 \times \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8}\right) = 1$$

所以

$$\eta = \frac{\frac{7}{4}}{1 \times \log_2 4} = \frac{7}{8}$$

$$R_d = 1 - \eta = 1 - \frac{7}{8} = \frac{1}{8} \quad \text{比特}$$

显然，编码后还有$\frac{1}{8}$比特的冗余度，没有达到 $\overline{N}$ 的最低限。

如果取 $A=\{0,1\}$，$n=2$，那么可以编成如下等长码：

$$u_1 = 00 \qquad u_3 = 10$$

$$u_2 = 01 \qquad u_4 = 11$$

此时

$$\overline{N} = 2 \times \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8}\right) = 2$$

$$\eta = \frac{\frac{7}{4}}{2 \times \log_2 2} = \frac{7}{8}$$

$$R_d = 1 - \frac{7}{8} = \frac{1}{8}$$

同样有$\frac{1}{8}$的冗余度。

上例中的两种编码方法，其特点是码字长度均相等，这种码叫等长码。显然此例中的两种等长码均没有达到最低限。怎样才能使信源编码达到最低限呢？再看下例的编码方法。仍选 $A=\{0,1\}$，$n=2$，作为编码字符集。在这种编码中，不用等长码，而是采用下面的原则来编码，即 P_i 大的消息编短码，P_i 小的消息编长码。

例

$$u_1 : 0$$

$$u_2 : 10$$

$$u_3 : 110$$

$$u_4 : 111$$

可计算出平均码长

$$\overline{N} = 1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + 3 \times \frac{1}{8} = \frac{7}{4}$$

其效率

$$\eta = \frac{\frac{7}{4}}{\frac{7}{4} \times \log_2 2} = 1$$

冗余度 $$R_d = 0$$

由此可见，这种编码法的码字平均长度达到了最低限。这说明用变长编码法可达到较高的效率。采用这种编码方法，信源中的消息与码字是一一对应的，因而译码时也是准确无误的。在编、译码过程中并不损失任何信息。它是一种信息保持编码法。

5.4.2 三种常用的统计编码法

变长编码是统计编码中最为主要的一种方法。变长编码的目标就是使平均码长达到低限，也就是使 $\overline{N}$ 最优，但是，这种最优必须在一定的限制下进行。编码的基本限制就是码字要有单义性和非续长性。

单义性代码是指任意一个有限长的码字序列只能被分割成一个一个的码字，而任何其他分割方法都会产生一些不属于码字集合中的码字。符合这个条件的代码就叫单义代码。

非续长代码是指任意一个码字都不是其他码字的续长。换句话说，就是码字集合中的任意一个码字都不是由其中一个码字在后面添上一些码元构成的。很容易看出非续长代码一定是单义的，但是，单义代码却不一定是非续长的。

例如，在表 5-3 中，列出四种代码，对码 I 来说，如果在收端收到 0 就无法判断是 u_1 还是 u_2，因此，在收端不能正确译码，显然码 I 缺乏单义可译性。对码 Ⅱ 来说，也有重要缺陷，例如，发端送 u_1u_1 这样一个序列，其码字将是 00，但是在收端既可判做 u_1u_1，也可以判做 u_3，所以这种码也缺乏单义可译性。而且，可以看出 u_3 的代码 00 是在 u_1 的代码 0 后面又加上一个 0 得到的，因此，又是可续长的。显然，码 Ⅱ 也不能使用。再看一下码 Ⅲ，码 Ⅲ 既具备单义可译性又是非续长的码，它是可用的。码 Ⅳ 也具有单义可译性，但是却缺乏非续长性。例如，当收到 0 时，不能立刻判断它是 u_1，必须等第二个码字出现，如果第二个码字是 0 则可以判断第一个码是 u_1，当第二个码字是 1 时，又无法判断了，还必须等待下一个码字出现才有可能判断，所以，续长码是无法及时译码的。

表 5-3 四种代码表

信源	概率	码 I	码 Ⅱ	码 Ⅲ	码 Ⅳ
u_1	$\frac{1}{2}$	0	0	0	0
u_2	$\frac{1}{4}$	0	1	10	01
u_3	$\frac{1}{8}$	1	00	110	011
u_4	$\frac{1}{8}$	10	11	111	0111

从上面的例子可知，使 $\overline{N}$ 最短的码只是在单义可译性和非续长性的约束下才有意义。至于变长码的存在定理以及 $\overline{N}$ 的最低限是否存在等问题，在信息论中都有详细的定理加以证明及讨论，在此不加赘述了。

最为常用的变长编码方法是霍夫曼(Huffman)码和仙农-费诺(Shannon-Fano)码。下面详细地讨论一下这两种码的构成方法。

1. 霍夫曼码

霍夫曼变长编码法能得到一组最优的变长码。设原始信源有 M 个消息，即

$$X=\begin{Bmatrix} u_1 & u_2 & u_3 & \cdots & u_M \\ P_1 & P_2 & P_3 & \cdots & P_M \end{Bmatrix} \tag{5-28}$$

可用下述步骤编出霍夫曼码：

第一步，把信源 $\boldsymbol{X}$ 中的消息按出现的概率从大到小的顺序排列，即 $P_1 \geqslant P_2 \geqslant P_3 \geqslant \cdots \geqslant P_M$。

第二步，把最后两个出现概率最小的消息合并成一个消息，从而使信源的消息数减少一个，并同时再次将信源中的消息的概率从大到小排列一次，得

$$\boldsymbol{X}_1 = \begin{Bmatrix} u'_1 & u'_2 & u'_3 & \cdots & u'_{M-1} \\ P'_1 & P'_2 & P'_3 & \cdots & P'_{M-1} \end{Bmatrix} \tag{5-29}$$

第三步，重复上述步骤，直到信源最后为 $\boldsymbol{X}^0$ 形式为止。这里 $\boldsymbol{X}^0$ 有如下形式：

$$\boldsymbol{X}^0 = \begin{Bmatrix} u^0_1 & u^0_2 \\ P^0_1 & P^0_2 \end{Bmatrix} \tag{5-30}$$

第四步，将被合并的消息分别赋以 1 和 0 或 0 和 1。对最后的 X^0 也对 u^0_1 和 u^0_2 对应地赋以 1 和 0 或 0 和 1。

通过上述步骤就可以构成最优变长码（霍夫曼码）。下面举例说明具体构成方法。

例　求下述信源的霍夫曼码：

$$\boldsymbol{X} = \begin{Bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ 0.25 & 0.25 & 0.20 & 0.15 & 0.10 & 0.05 \end{Bmatrix}$$

由上述步骤，做一个新的信源：

$$\boldsymbol{X}' = \begin{Bmatrix} u'_1 & u'_2 & u'_3 & u'_4 & u'_5 \\ 0.25 & 0.25 & 0.20 & 0.15 & 0.15 \end{Bmatrix}$$

这样可给 u_5 赋 0，u_6 赋 1，其中 $u'_5=(u_5+u_6)$。$\boldsymbol{X}'$ 中消息的概率大小顺序正好符合从大到小的规律，故不必重排。再做新的信源：

$$\boldsymbol{X}'' = \begin{Bmatrix} u''_1 & u''_2 & u''_3 & u''_4 \\ 0.25 & 0.25 & 0.20 & 0.30 \end{Bmatrix}$$

重排得

$$\boldsymbol{X}'' = \begin{Bmatrix} u''_1 & u''_2 & u''_3 & u''_4 \\ 0.30 & 0.25 & 0.25 & 0.20 \end{Bmatrix}$$

将 u'_4 赋 0，u'_5 赋 1。将 u''_3 和 u''_4 合并构成新的信源：

$$\boldsymbol{X}''' = \begin{Bmatrix} u'''_1 & u'''_2 & u'''_3 \\ 0.30 & 0.25 & 0.45 \end{Bmatrix}$$

重排得

$$\boldsymbol{X}''' = \begin{Bmatrix} u'''_1 & u'''_2 & u'''_3 \\ 0.45 & 0.30 & 0.25 \end{Bmatrix}$$

将 u''_3 赋 0，u''_4 赋 1。最后则

$$\boldsymbol{X}^0 = \begin{Bmatrix} u^0_1 & u^0_2 \\ 0.45 & 0.55 \end{Bmatrix}$$

重排得

$$\boldsymbol{X}^0 = \begin{Bmatrix} u^0_1 & u^0_2 \\ 0.55 & 0.45 \end{Bmatrix}$$

u'''_2 赋 0，u'''_3 赋 1。最后 u^0_1 赋 0，u^0_2 赋 1。编码结果可总结于表 5-4 中。编码图如图 5-18 所

示。

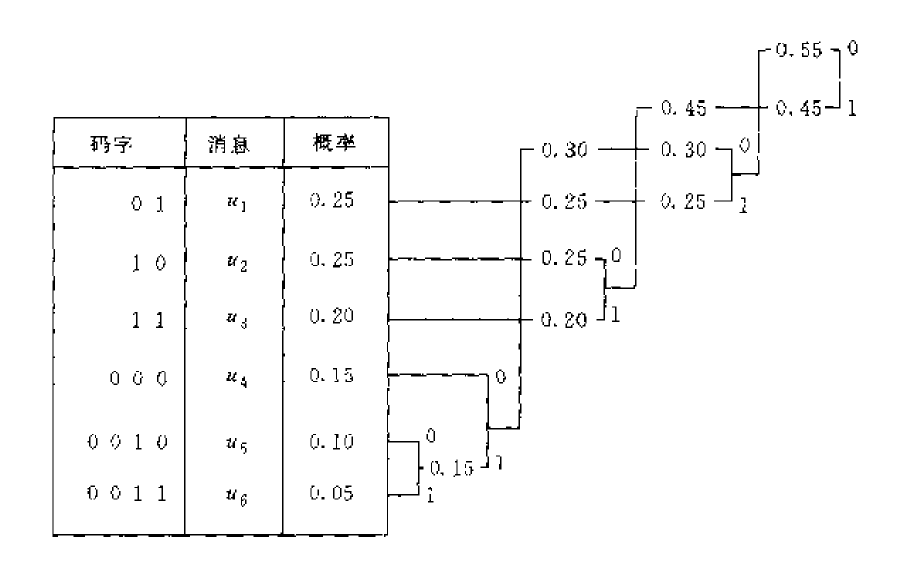


图 5-18 信源 X 的霍夫曼编码图

如对合并的消息赋以 1,0 值,则会得到如表 5-5 所示的另外一组码。

表 5-4 信源 X 的霍夫曼编码表

码 字	消 息	概 率
0 1	u_1	0.25
1 0	u_2	0.25
1 1	u_3	0.20
0 0 0	u_4	0.15
0 0 1 0	u_5	0.10
0 0 1 1	u_6	0.05

表 5-5 信源 X 的另一组霍夫曼编码表

码 字	消 息	概 率
1 0	u_1	0.25
0 1	u_2	0.25
0 0	u_3	0.20
1 1 1	u_4	0.15
1 1 0 1	u_5	0.10
1 1 0 0	u_6	0.05

下面计算一下信源的熵,平均码长,效率及冗余度。

$$H(\boldsymbol{X}) = -\sum_{i=1}^{6} P_i \log_2 P_i$$

$$= -0.25\log_2 0.25 - 0.25\log_2 0.25 - 0.20\log_2 0.20$$

$$- 0.15\log_2 0.15 - 0.10\log_2 0.10 - 0.05\log_2 0.05 = 2.42$$

$$\overline{N} = 2 \times 0.25 + 2 \times 0.25 + 2 \times 0.20 + 3 \times 0.15 + 4 \times 0.10 + 4 \times 0.05 = 2.45$$

$$\eta = \frac{H(\boldsymbol{X})}{\overline{N}\log_2 n} = \frac{2.42}{2.45\log_2 2} = 0.98 = 98\%$$

$$R_d = 1 - \eta = 1 - 98\% = 2\%$$

所以,对于信源 $\boldsymbol{X}$ 的霍夫曼码的编码效率为 98%,尚有 2% 的冗余度。

2. 仙农-费诺码

另外一种常用的变长编码是仙农-费诺码。这种码有时也可以得到最优编码性能。它的编码准则要符合非续长条件,在码字中 1 和 0 是独立的,而且是(或差不多是)等概率的。这样的准则一方面可保证无需用间隔来区分码字,同时又保证每传输 1 位码就有 1bit 的信息量。

仙农-费诺码的编码程序可由下述几个步骤来完成:

第一步 设信源 $\boldsymbol{X}$ 有非递增的概率分布:

$$X=\begin{Bmatrix} u_1 & u_2 & u_3 & \cdots & u_M \\ P_1 & P_2 & P_3 & \cdots & P_M \end{Bmatrix} \tag{5-31}$$

其中 $P_1 \geqslant P_2 \geqslant P_3 \geqslant \cdots \geqslant P_M$。把 X 分成两个子集合，得：

$$X_1=\begin{Bmatrix} u_1 & u_2 & u_3 & \cdots & u_k \\ P_1 & P_2 & P_3 & \cdots & P_k \end{Bmatrix} \tag{5-32}$$

$$X_2=\begin{Bmatrix} u_{k+1} & u_{k+2} & u_{k+3} & \cdots & u_M \\ P_{k+1} & P_{k+2} & P_{k+3} & \cdots & P_M \end{Bmatrix} \tag{5-33}$$

并且保证

$$\sum_{i=1}^{k} P_i \approx \sum_{i=k+1}^{M} P_i \tag{5-34}$$

成立或差不多成立。

第二步　给两个子集中的消息赋值，X_1 赋 1，X_2 赋 0，或给 X_1 赋 0，X_2 赋 1。

第三步　重复第一步骤，将两个子集 X_1，X_2 再细分为两个子集，并且也同样使两个小子集里消息的概率之和相等或近似相等。然后，重复第二步骤赋值。以这样的步骤重复下去，直到每个子集内只包含一个消息为止。对每个消息所赋过的值依次排列出来就可以构成仙农-费诺码

下面举例说明仙农-费诺码的具体构成方法。

例　设有信源

$$X=\begin{Bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \end{Bmatrix}$$

其编码流程图如图 5-19 所示。编码表如表 5-6 所示。如果对各子集赋以另外一种值，即 1，0，那么，同样会得到另一种编码结果，其编码表如表 5-7 所示。

码字	消息	概率				
0 0	u_1	1/4	0	0		
0 1	u_2	1/4		1		
1 0 0	u_3	1/8			0	
1 0 1	u_4	1/8		0	1	
1 1 0 0	u_5	1/16	1			
1 1 0 1	u_6	1/16		1	0	
1 1 1 0	u_7	1/16			1	0
1 1 1 1	u_8	1/16				1

图 5-19　仙农-费诺码编码流程图

下面计算一下仙农-费诺码的平均码长，效率及冗余度。信源的熵可计算于下：

$$\begin{aligned} H(X) &= -\frac{1}{4}\log_2\frac{1}{4}-\frac{1}{4}\log_2\frac{1}{4}-\frac{1}{8}\log_2\frac{1}{8}-\frac{1}{8}\log_2\frac{1}{8} \\ &\quad -\frac{1}{16}\log_2\frac{1}{16}-\frac{1}{16}\log_2\frac{1}{16}-\frac{1}{16}\log_2\frac{1}{16}-\frac{1}{16}\log_2\frac{1}{16} \\ &= 2\frac{3}{4}\ \text{比特/消息} \end{aligned}$$

表 5-6　仙农-费诺码编码表

消息	概率	码字
u_1	$\frac{1}{4}$	0 0
u_2	$\frac{1}{4}$	0 1
u_3	$\frac{1}{8}$	1 0 0
u_4	$\frac{1}{8}$	1 0 1
u_5	$\frac{1}{16}$	1 1 0 0
u_6	$\frac{1}{16}$	1 1 0 1
u_7	$\frac{1}{16}$	1 1 1 0
u_8	$\frac{1}{16}$	1 1 1 1

表 5-7　另一种仙农-费诺码编码表

消息	概率	码字
u_1	$\frac{1}{4}$	1 1
u_2	$\frac{1}{4}$	1 0
u_3	$\frac{1}{8}$	0 1 1
u_4	$\frac{1}{8}$	0 1 0
u_5	$\frac{1}{16}$	0 0 1 1
u_6	$\frac{1}{16}$	0 0 1 0
u_7	$\frac{1}{16}$	0 0 0 1
u_8	$\frac{1}{16}$	0 0 0 0

平均码长为

$$\overline{N}=\sum_{i=1}^{8}P_iN_i$$
$$=\frac{1}{4}\times 2+\frac{1}{4}\times 2+\frac{1}{8}\times 3+\frac{1}{8}\times 3+\frac{1}{16}\times 4$$
$$+\frac{1}{16}\times 4+\frac{1}{16}\times 4+\frac{1}{16}\times 4=2\frac{3}{4}$$

显然，$\eta=1$，$R_d=0$。效率已达到 100%。这一结果并不奇怪，对于仙农-费诺码来说，如果满足下式：

$$P(u_i)=2^{-N_i} \tag{5-35}$$

且

$$\sum_{i=1}^{M}2^{-N_i}=1 \tag{5-36}$$

就会使编码效率达到 100%。式中的 $P(u_i)$为消息 u_i 出现的概率，N_i 是码字的长度。如果不满足上述条件就不会有 100%的效率，如下例中图 5-20 所示。

码字	消息	概率
0	u_1	0.49
1 0 0	u_2	0.14
1 0 1	u_3	0.14
1 1 0 0	u_4	0.07
1 1 0 1	u_5	0.07
1 1 1 0	u_6	0.04
1 1 1 1 0	u_7	0.02
1 1 1 1 1 0	u_8	0.02
1 1 1 1 1 1	u_9	0.01

图 5-20　编码流程图

例　设有一信源：

$$X=\begin{Bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ 0.49 & 0.14 & 0.14 & 0.07 & 0.07 & 0.04 & 0.02 & 0.02 & 0.01 \end{Bmatrix}$$

编码流程及形成的码字如图 5-20 所示。可以计算出信源的熵为

$$H(X) = 2.313$$

平均码长 $\overline{N}=2.33$

效率 $\eta=0.993$

冗余度 $R_d=0.007$

由此例可见，由于信源不满足式(5-35)和(5-36)的条件，编码效率不能达到100%。然而从结果上看，它仍然是一种相当好的编码。

3. 霍夫曼-仙农-费诺码

我们把霍夫曼-仙农-费诺码简称HSF码。这种码兼有霍夫曼和仙农-费诺码的特点，即具有最小冗余度及有数值序列的特点。也就是说，从概率大到概率小的码字来看它的数值大小是递增的(这指的是将二进制码字翻译成十进制数的值)。这一特点正好被利用来减少翻译表的大小，缩短编、译码的时间。这样HSF码就兼有两种码的优点。HSF码的编码方法将用下面的例子加以说明。

例 设有一个信源

$$X=\begin{Bmatrix} 2 & 9 & 4 & 0 & A & 8 & 3 & 7 & 1 & 5 & B & 6 \\ 0.226 & 0.165 & 0.135 & 0.120 & 0.079 & 0.063 & 0.054 & 0.041 & 0.038 & 0.034 & 0.030 & 0.015 \end{Bmatrix}$$

这个信源共有12个消息(或字符)，它们出现的概率不等。首先编成霍夫曼码：

$$X_H=\{10,000,010,011,0010,1101,1111,00110,00111,11100,11101\}$$

由编码可见，码字长度分别为2,3,4,5，因此，可以定义一个编码索引I，

$$I = W_1W_2W_3W_4W_5 \tag{5-37}$$

其中W_i代表字长为i的码字的个数。在这个例子中，字长为1的码字没有，所以$W_1=0$，字长为2的码字有一个，所以$W_2=1$。以此类推可得：

$$I = 01344$$

根据I可以构造HSF码，也就是用SF码的构码方法编码，但它的字长和个数要受I的约束。显然，HSF码的平均码长$\overline{N}$与霍夫曼码一样。如果用编码树来表示，如图5-21所示。编码表示于表5-8中(为清楚起见，把霍夫曼码和HSF码均列在表5-8中，以便对照比较)。从表中可见，霍夫曼码没有数值序列特性，而HSF码的数值是从0到(2^k-1)递增的，其中k是最后一个码字的码位数。表中把HSF码及其数值列于第5、第6列。依照HSF码的特性可以构造出较短的翻译表，从而可以减少编、译码的时间。

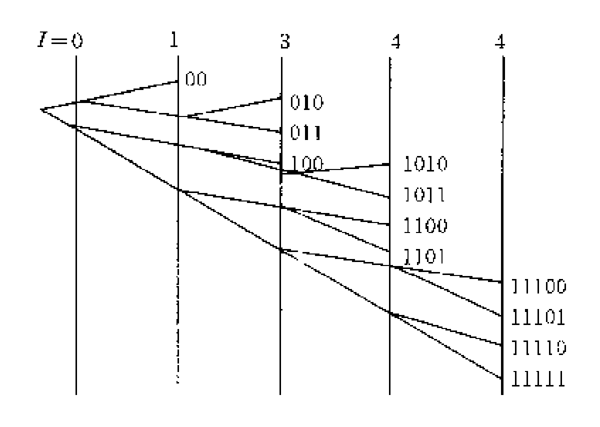


图5-21 HSF码编码树

表 5-8 HSF 编码表

消息字符	概　率	概率顺序	霍夫曼码	HSF 码	HSF 码值
2	0.226	0	1 0	0 0	0
9	0.165	1	0 0 0	0 1 0	2
4	0.135	2	0 1 0	0 1 1	3
0	0.120	3	0 1 1	1 0 0	4
A	0.079	4	0 0 1 0	1 0 1 0	10
8	0.063	5	1 1 0 0	1 0 1 1	11
3	0.054	6	1 1 0 1	1 1 0 0	12
7	0.041	7	1 1 1 1	1 1 0 1	13
1	0.038	8	0 0 1 1 0	1 1 1 0 0	28
5	0.034	9	0 0 1 1 1	1 1 1 0 1	29
B	0.030	10	1 1 1 0 0	1 1 1 1 0	30
6	0.015	11	1 1 1 0 1	1 1 1 1 1	31

建立 HSF 翻译表有 3 个：

(1) A 表(表 5-9)

表 5-9 A 表

3	8	0	6	2	9	11	7	5	1	4	10

这个表是这样建立的，表的顺序从左到右按消息字符的值从小到大自然顺序放置。每个位置中的数字则是此字符出现的概率的顺序数(概率是从大到小排列)。例如，消息字符 0 的概率顺序是 3，消息字符 3 的概率顺序是 6，A 的概率顺序是 4 等等。分别在表 A 的第 1 格填上 3，第 4 格填上 6，第 11 格填上 4 等等。A 表是编码表。

(2) B 表(表 5-10)

表 5-10 B 表

2	9	4	0	A	8	3	7	1	5	B	6

此表是译码表。它的意义与 A 表正好相反。表中位置顺序是消息概率顺序从小到大排列。格中的字符就是消息字符。例如消息概率序数为 0 的正好是字符 2，消息概率序数为 5 的正好是字符 8 等等。

(3) C 表(表 5-11)

表 5-11 C 表

极限值	基值	范围值
0 0 (0)	0	0
1 0 0 (4)	1	3
1 1 0 1 (13)	6	7
1 1 1 1 1 (31)	20	11

此表既是编码表又是译码表。表中共有 3 列。第 1 列是极限值，它指明每一组相同字长的码字中的最大数值，括弧中指的是对应的十进制数值。第 2 列是基值。每一组相同字长的码字中有一个基值，如字长为 2 的这一组码里基值为 0，字长 4 的这一组码里基值为 6 等等。这一

组里的码字的值(即表 5-8 中的 HSF 码值)减去这一基值就会得到消息序列的概率顺序值。而基值与范围值相加就正好等于极限值。第 3 列是范围值,它与每组码字中的最大概率顺序一致。例如,在 5 位字长的一组码字中概率顺序最大数是 11,则范围值为 11。在 4 位字长一组中概率顺序最大数是 7,则范围数值为 7 等等。

根据以上三个表就可以进行 HSF 编码了。例如,当信源送出消息字符 8 时,由表 A,在第 8 位置上查出其概率顺序为 5。于是,将它与范围值逐次比较,发现它小于 7,这样就可确定它在 C 表第三行内,其字长为 4,基值为 6,把 5 与 6 相加便得 11,把 11 翻成二进制码就是 1011。这就是消息字符 8 的 HSF 码。这种编码方法快速简便。

译码同样比较简单。假定某个比特序列是 110000011…。第一步比较头二个比特(11)(3)与第一个极限值(00)(0)相比,发现(11)>(00),则再比较头三个比特(110)(6),它与第二个极限值比较发现(110)>(100),则再比较头四个比特(1100)(12),(1100)与第三个极限值比较,发现(1100)<(1101),这样,就可以判断 1100 这一码字在 C 表的第三行。因此,字长为 4,基值为 6,于是用 1100 的数值减去基值 6 得 6。然后再查 B 表第 6 位置上的数字为 3,这就是要译出的消息字符 3。

总的来说,HSF 码兼有霍夫曼码与仙农-费诺码的优点,因此,在码字长度及编译码速度上均比较优越。

经过上面的讨论来看,统计编码是一种高效编码法。但是,它也有一些缺点,其一是它的码字不是等长的,在使用中需要用数据缓存单元收集可变比特率的代码,并以较慢的平均速率传输,这对使用来说不太方便。其次,这几种码都缺乏构造性,也就是说它们都不能用数学方法建立一一对应关系,而只能通过查表的方法来实现对应关系。如果消息数目太多,表就会很大,所需要的存储器也越多,相对的设备也就越复杂。再有一个难题就是在编码过程中应知道每种消息可能出现的概率。在图像编码中就是要知道每种图像信息出现的概率。实际上这种概率很难估计或测量,如果不能恰当地利用这种概率便会使编码性能明显下降,因此,目前使用的方法多需进一步改进。

在 ITU-T 建议的彩色图像编码标准中的编码表如表 5-12 所示。

表 5-12 AC 系数 Huffman 码表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
0/0	4	1010	2	00
0/1	2	00	2	01
0/2	2	01	3	100
0/3	3	100	4	1010
0/4	4	1011	5	11000
0/5	5	11010	5	11001
0/6	7	1111000	6	111000
0/7	8	11111000	7	1111000
0/8	10	1111110110	9	111110100
0/9	16	1111111110000010	10	1111110110
0/A	16	1111111110000011	12	111111110100
1/1	4	1100	4	1011

续表

游程/尺寸	亮度AC系数		色度AC系数	
	码长	码字	码长	码字
1/2	5	11011	6	111001
1/3	7	1111001	8	11110110
1/4	9	111110110	9	111110101
1/5	11	11111110110	11	11111110110
1/6	16	1111111110000100	12	111111110101
1/7	16	1111111110000101	16	1111111110001000
1/8	16	1111111110000110	16	1111111110001001
1/9	16	1111111110000111	16	1111111110001010
1/A	16	1111111110001000	16	1111111110001011
2/1	5	11100	5	11010
2/2	8	11111001	8	11110111
2/3	10	1111110111	10	1111110111
2/4	12	111111110100	12	111111110110
2/5	16	1111111110001001	15	111111111000010
2/6	16	1111111110001010	16	1111111110001100
2/7	16	1111111110001011	16	1111111110001110
2/8	16	1111111110001100	16	1111111110001110
2/9	16	1111111110001101	16	1111111110001111
2/A	16	1111111110001110	16	1111111110010000
3/1	6	111010	5	11011
3/2	9	111110111	8	11111000
3/3	12	111111110101	10	1111111000
3/4	16	1111111110001111	12	111111110111
3/5	16	1111111110010000	16	1111111110010001
3/6	16	1111111110010001	16	1111111110010010
3/7	16	1111111110010010	16	1111111110010011
3/8	16	1111111110010011	16	1111111110010100
3/9	16	1111111110010100	16	1111111110010101
3/A	16	1111111110010101	16	1111111110010110
4/1	6	111011	6	111010
4/2	10	1111111000	9	111110110
4/3	16	1111111110010110	16	1111111110010111
4/4	16	1111111110010111	16	1111111110011000
4/5	16	1111111110011000	16	1111111110011001
4/6	16	1111111110011001	16	1111111110011010
4/7	16	1111111110011010	16	1111111110011011
4/8	16	1111111110011011	16	1111111110011100
4/9	16	1111111110011100	16	1111111110011101
4/A	16	1111111110011101	16	1111111110011110
5/1	7	1111010	6	111011
5/2	11	11111110111	10	1111111001
5/3	16	1111111110011110	16	1111111110011111
5/4	16	1111111110011111	16	1111111110100000
5/5	16	1111111110100000	16	1111111110100001
5/6	16	1111111110100001	16	1111111110100010
5/7	16	1111111110100010	16	1111111110100011
5/8	16	1111111110100011	16	1111111110100100
5/9	16	1111111110100100	16	1111111110100101

续表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
5/A	16	1111111110100101	16	1111111110100110
6/1	7	1111011	7	1111001
6/2	12	111111110110	11	11111110111
6/3	16	1111111110100110	16	1111111110100111
6/4	16	1111111110100111	16	1111111110101000
6/5	16	1111111110101000	16	1111111110101001
6/6	16	1111111110101001	16	1111111110101010
6/7	16	1111111110101010	16	1111111110101011
6/8	16	1111111110101011	16	1111111110101100
6/9	16	1111111110101100	16	1111111110101101
6/A	16	1111111110101101	16	1111111110101110
7/1	8	11111010	7	1111010
7/2	12	111111110111	11	11111111000
7/3	16	1111111110101110	16	1111111110101111
7/4	16	1111111110101111	16	1111111110110000
7/5	16	1111111110110000	16	1111111110110001
7/6	16	1111111110110001	16	1111111110110010
7/7	16	1111111110110010	16	1111111110110011
7/8	16	1111111110110011	16	1111111110110100
7/9	16	1111111110110100	16	1111111110110101
7/A	16	1111111110110101	16	1111111110110110
8/1	9	111111000	8	11111001
8/2	15	111111111000000	16	1111111110110111
8/3	16	1111111110110110	16	1111111110111000
8/4	16	1111111110110111	16	1111111110111001
8/5	16	1111111110111000	16	1111111110111010
8/6	16	1111111110111001	16	1111111110111011
8/7	16	1111111110111010	16	1111111110111100
8/8	16	1111111110111011	16	1111111110111101
8/9	16	1111111110111100	16	1111111110111110
8/A	16	1111111110111101	16	1111111110111111
9/1	9	111111001	9	111110111
9/2	16	1111111110111110	16	1111111111000000
9/3	16	1111111110111111	16	1111111111000001
9/4	16	1111111111000000	16	1111111111000010
9/5	16	1111111111000001	16	1111111111000011
9/6	16	1111111111000010	16	1111111111000100
9/7	16	1111111111000011	16	1111111111000101
9/8	16	1111111111000100	16	1111111111000110
9/9	16	1111111111000101	16	1111111111000111
9/A	16	1111111111000110	16	1111111111001000
A/1	9	111111010	9	111111000
A/2	16	1111111111000111	16	1111111111001001
A/3	16	1111111111001000	16	1111111111001010
A/4	16	1111111111001001	16	1111111111001011
A/5	16	1111111111001010	16	1111111111001100
A/6	16	1111111111001011	16	1111111111001101
A/7	16	1111111111001100	16	1111111111001110

续表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
A/8	16	1111111111001101	16	1111111111001111
A/9	16	1111111111001110	16	1111111111010000
A/A	16	1111111111001111	16	1111111111010001
B/1	10	1111111001	9	111111001
B/2	16	1111111111010000	16	1111111111010010
B/3	16	1111111111010001	16	1111111111010011
B/4	16	1111111111010010	16	1111111111010100
B/5	16	1111111111010011	16	1111111111010101
B/6	16	1111111111010100	16	1111111111010110
B/7	16	1111111111010101	16	1111111111010111
B/8	16	1111111111010110	16	1111111111011000
B/9	16	1111111111010111	16	1111111111011001
B/A	16	1111111111011000	16	1111111111011010
C/1	10	1111111010	9	111111010
C/2	16	1111111111011001	16	1111111111011011
C/3	16	1111111111011010	16	1111111111011100
C/4	16	1111111111011011	16	1111111111011101
C/5	16	1111111111011100	16	1111111111011110
C/6	16	1111111111011101	16	1111111111011111
C/7	16	1111111111011110	16	1111111111100000
C/8	16	1111111111011111	16	1111111111100001
C/9	16	1111111111100000	16	1111111111100010
C/A	16	1111111111100001	16	1111111111100011
D/1	11	11111111000	11	11111111001
D/2	16	1111111111100010	16	1111111111100100
D/3	16	1111111111100011	16	1111111111100101
D/4	16	1111111111100100	16	1111111111100110
D/5	16	1111111111100101	16	1111111111100111
D/6	16	1111111111100110	16	1111111111101000
D/7	16	1111111111100111	16	1111111111101001
D/8	16	1111111111101000	16	1111111111101010
D/9	16	1111111111101001	16	1111111111101011
D/A	16	1111111111101010	16	1111111111101100
E/1	16	1111111111101011	14	11111111100000
E/2	16	1111111111101100	16	1111111111101101
E/3	16	1111111111101101	16	1111111111101110
E/4	16	1111111111101110	16	1111111111101111
E/5	16	1111111111101111	16	1111111111110000
E/6	16	1111111111110000	16	11111111111110001
E/7	16	1111111111110001	16	1111111111110010
E/8	16	1111111111110010	16	1111111111110011
E/9	16	1111111111110011	16	1111111111110100
E/A	16	1111111111110100	16	1111111111110101
F/0	11	11111111001	10	1111111010
F/1	16	1111111111110101	15	111111111000011
F/2	16	1111111111110110	16	1111111111110100
F/3	16	1111111111110111	16	1111111111110111
F/4	16	1111111111111000	16	1111111111111000

续表

游程/尺寸	亮度 AC 系数		色度 AC 系数	
	码长	码字	码长	码字
F/5	16	1111111111111001	16	1111111111111001
F/6	16	1111111111111010	16	1111111111111010
F/7	16	1111111111111011	16	1111111111111011
F/8	16	1111111111111100	16	1111111111111100
F/9	16	1111111111111101	16	1111111111111101
F/A	16	1111111111111110	16	1111111111111110

5.5 预测编码

预测编码法是一种设备简单质量较佳的高效编码法。预测编码方法主要有二种。一种是 ΔM(Delta Modulation)或 DM 编码法，另一种是 DPCM(Differential Pulse Code Modulation)编码法。本节主要介绍这两种方法的原理及其在图像编码中的应用。

5.5.1 预测编码的基本原理

预测编码的基本原理如图 5-22 所示。假设有一个平均值为零，均方根值为 σ 的平稳信号 $X(t)$ 在时刻 $t_1, t_2, \cdots, t_n$ 被取样，而且其相应的样值为 $x_1, x_2, \cdots, x_n$。在图 5-22(a)的编码原理图中，x_i 是下一个样值。根据前面出现的 n 个样值，可以得到 x_i 的预测值 $\hat{x}_i$：

$$\hat{x}_i = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \cdots + \alpha_n x_n \tag{5-38}$$

式中 $x_1, x_2, \cdots, x_n$ 是 x_i 的前 n 个样值。$\alpha_1, \alpha_2, \cdots, \alpha_n$ 是预测参数。设 e_i 为 x_i 与 $\hat{x}_i$ 之间的误差值，则

$$e_i = x_i - \hat{x}_i \tag{5-39}$$

预测编码就是要对误差 e_i 进行编码，而不是对样值直接编码。那么，对误差编码果真可以压缩数据率吗？下面先定性地分析一下其可能性。

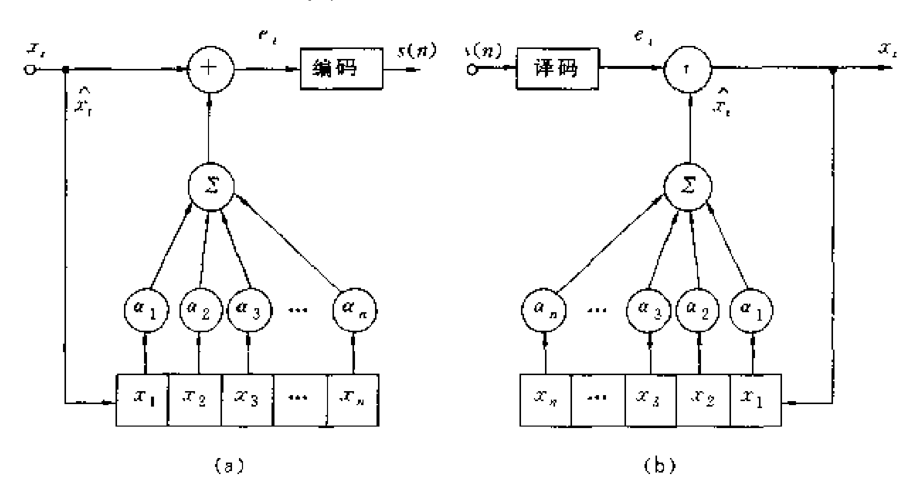


图 5-22 预测编码原理

假如直接对样值 x 编码，那么正如前面谈到的那样，代码平均长度有一个下限 $\overline{N}_{\min}$，这个下限就是信源的熵 $H(X)$，即

$$\overline{N}_{\min} = H(X) = -\sum_i P(i)\lg P(i) \tag{5-40}$$

同样道理，如果对误差信号进行编码，那么，它也应该有一个下限，设为 $H(E)$。显然，预测编码可以压缩数码率的条件是

$$H(E) < H(X) \tag{5-41}$$

熵是概率分布的函数，分布越均匀熵越大。熵值大，则其平均码长之下限必然会加大，码率就会增高。反之，分布越集中熵值越小，而其平均码长之下限就会越短，码率就会降低。如果预测比较准确，那么误差就会集中于不大的数值内，从而使 $H(E)$ 小于 $H(X)$。由于图像信号中样值的高度相关性，使得相邻样值之间的差别总是十分微小的，所以其差值分布十分集中。预测前后的概率分布情况如图 5-23 所示。

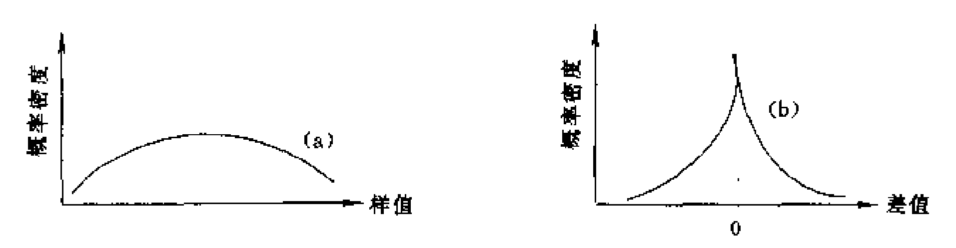


图 5-23 预测前后的概率密度分布示意图

(a)图像信号概率密度分布， (b)差值信号概率密度分布

在第 2 章关于图像信号性质的讨论中可知，帧内像素相关系数在 0.85 左右，帧间相关系数在 0.95 左右。由此可见，图像像素间的相关性是很大的，其压缩潜力也是很大的。由上面的定性分析可知，预测编码是可以压缩码率的。

一般情况，使用线性预测器，预测值与前面的 n 个已出现样值的关系如式(5-38)所示。线性预测的关键一步在于预测系数 α_i 的求解。预测误差信号是一个随机变量，它的均方误差为 σ_i^2，

$$\sigma_i^2 = E[(x_i - \hat{x}_i)^2] \tag{5-42}$$

这里 $E[\ ]$表示数学期望。通常把均方误差最小的预测称为最佳预测。通过最小均方误差准则可求解预测系数，即

$$\frac{\partial E[(x_i - \hat{x}_i)^2]}{\partial \alpha_j} = 0 \qquad j = 1,2,\cdots,n \tag{5-43}$$

将式(5-38)代入，则

$$\begin{aligned} &\frac{\partial E[(x_i - \hat{x}_i)^2]}{\partial \alpha_j} \\ &= \frac{\partial E[(x_i - (\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n))^2]}{\partial \alpha_j} \\ &= -2E[(x_i - (\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n))x_j] \end{aligned} \tag{5-44}$$

为求极小值可令式(5-44)等于 0，即

$$E[(x_i - (\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n))x_j] = 0$$

或

$$E[(x_i - \hat{x}_i)x_j] = 0 \qquad j = 1,2,\cdots,n \tag{5-45}$$

因为信号 x 是平稳的随机过程，并且均值为零，所以可将任意两个像素的协方差定义为

R_{ij}，

$$R_{ij} = E[x_i x_j] \quad (5\text{-}46)$$

展开式(5-45)得：

$$E[x_i x_j - \alpha_1 x_1 x_j - \alpha_2 x_2 x_j - \cdots - \alpha_n x_n x_j] = 0$$

令式中 $j=1,2,\cdots,n$；$i=0$ 则

$$\begin{cases} R_{01} = \alpha_1 R_{11} + \alpha_2 R_{21} + \cdots + \alpha_n R_{n1} \\ R_{02} = \alpha_1 R_{12} + \alpha_2 R_{22} + \cdots + \alpha_n R_{n2} \\ \qquad\cdots \\ R_{0n} = \alpha_1 R_{1n} + \alpha_2 R_{2n} + \cdots + \alpha_n R_{nn} \end{cases} \quad (5\text{-}47)$$

这是一个 n 阶线性联立方程组，当协方差 R_{ij}都已知时，那么各个预测参数 α_i 是可以解出来的。

另外，由上面的讨论可知，如果 $\hat{x}_i$ 是 x_i 的最佳线性估计值，则

$$E[(x_i - \hat{x}_i)x_j] = 0 \qquad j = 1,2,\cdots,n$$

而其均方误差为

$$\begin{aligned} \sigma_i^2 &= E[(x_i - \hat{x}_i)^2] \\ &= E[(x_i - \hat{x}_i)(x_i - \hat{x}_i)] \\ &= E[(x_i - \hat{x}_i)x_i - (x_i - \hat{x}_i)\hat{x}_i] \\ &= E[(x_i - \hat{x}_i)x_i] - E[(x_i - \hat{x}_i)\hat{x}_i] \\ &= E[(x_i - \hat{x}_i)x_i] - E[(x_i - \hat{x}_i)(\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n)] \\ &= E[(x_i - \hat{x}_i)x_i] - E[\alpha_1(x_i - \hat{x}_i)x_1] - E[\alpha_2(x_i - \hat{x}_i)x_2] \\ &\quad - \cdots - E[\alpha_n(x_i - \hat{x}_i)x_n] = E[(x_i - \hat{x}_i)x_i] \end{aligned}$$

由此可得

$$\sigma_i^2 = E[(x_i - \hat{x}_i)x_i] \quad (5\text{-}48)$$

当 $i=0$ 时，则

$$\sigma_0^2 = E[x_0^2 - x_0\hat{x}_0] \quad (5\text{-}49)$$

将
$$\hat{x}_0 = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$$

代入式(5-49)，并引入协方差之定义，则

$$\sigma_0^2 = R_{00} - (\alpha_1 R_{01} + \alpha_2 R_{02} + \cdots + \alpha_n R_{0n}) \quad (5\text{-}50)$$

式中 R_{00}是原序列 X 的方差。由式(5-50)可见，误差序列的方差 σ_0^2 比原序列的方差确实要小。如果在形成估计时所用的取样值 n 无限制时，那么误差取样序列总可以是完全不相关的。如果取样序列是 r 阶马尔可夫序列，则在形成 x_i 的最佳估计中，只需采用 r 个取样值，而且得出的误差取样序列也会是不相关的。由于解除了样值间的相关性，也就解除了存在于相关性中的多余度。

对于图像编码，特别是电视信号编码，如果利用同一行的前 r 个样值进行预测，叫一维预测。如果同时利用前面几行的样值预测就叫二维预测。电视图像一般是一帧一帧连续发送的，那么可以利用前面若干帧进行预测，这时就是三维预测了。

对于电视信号来说，可认为它是一阶马尔可夫过程，这时只采用前值预测法便可以了。其误差值为

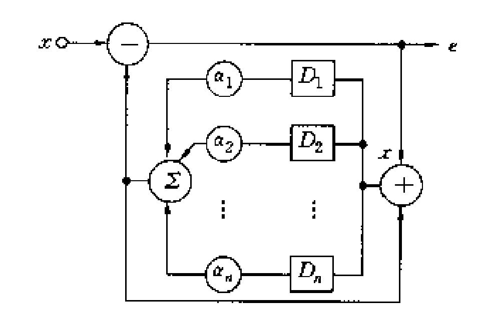
$$e_i = x_i - \hat{x}_i = x_i - \alpha_1 x_{i-1} \quad (5\text{-}51)$$

此时，电视信号取样序列的自相关函数近于指数形式，即 $e^{-\alpha t}$的形式。

线性预测的原理框图如图 5-24 所示，在图中，输入信号为 x，x 的最佳估计为 $\hat{x}$；$D_1,D_2,\cdots,D_n$ 为延迟单元；$\alpha_1,\alpha_2,\cdots,\alpha_n$ 为预测系数。如果采用前值预测，则

$$\hat{x}=\alpha_1 x_1 \tag{5-52}$$

前值预测原理框图如图 5-25 所示。



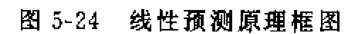


图 5-24　线性预测原理框图

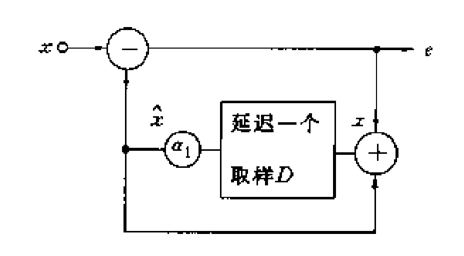


图 5-25　前值预测原理框图

如果如图 5-26 所示，利用前行的 x_2 及本行的前一个样值 x_1 来预测 x_0，预测器原理框图如图 5-27 所示。

以上便是预测编码的基本原理。目前，应用和实验研制的预测编码方法主要有两大类。一类是增量调制编码(ΔM)，另外一类就是差分脉冲编码调制编码。两类编码法中又都有各自的各种自适应方案。下面将较详细的介绍这两类预测编码法。

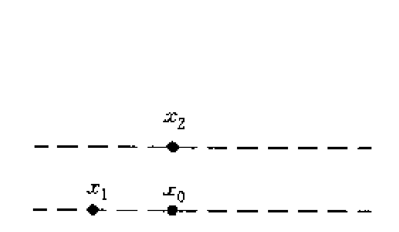


图 5-26　利用前一行样值 x_2 及本行前一个样值预测 x_0 的示意图

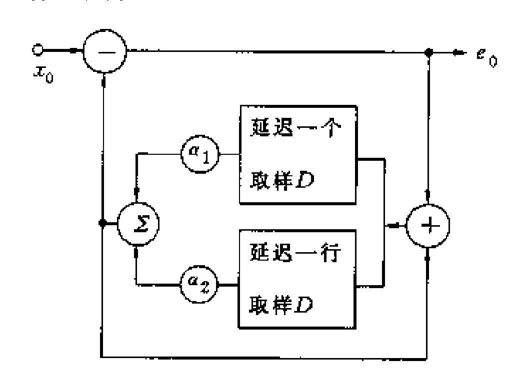


图 5-27　利用前一行样值 x_2 及本行前值 x_1 预测 x_0 的原理框图

5.5.2　ΔM(DM)编码

1. ΔM 编码的基本原理

ΔM 编码基本原理框图如图 5-28 所示，其中(a)为编码原理框图，(b)为译码原理框图。ΔM 编码器包括比较器、本地译码器和脉冲形成器三个部分。收端译码器比较简单，它只有一个与编码器中的本地译码一样的译码器及一个视频带宽的低通滤波器。

ΔM 编码器实际上就是 1bit 编码的预测编码器。它用一位码字来表示 $e(t)$，

$$e(t)=f(t)-\hat{f}(t) \tag{5-53}$$

式中 $f(t)$为输入视频信号，$\hat{f}(t)$是 $f(t)$的预测值。当差值 $e(t)$为一个正的增量时用"1"码来表示，当差值 $e(t)$为一个负的增量时用"0"码来表示。在收端，当译码器收到"1"时，信号则产生

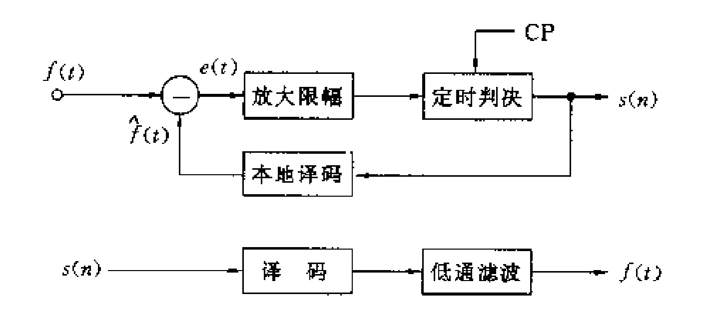


图 5-28 ΔM 编码、译码原理方框图

一个正跳变，收到“0”时，则信号电压产生一个负的跳变，由此即可实现译码。

根据上述原理，首先讨论一下译码电路。一般说来，译码器应具有下述三个功能：

1）收到“1”时，产生一个正斜变电压，当连续收到“1”时，则连续上升；

2）收到“0”时，产生一个负斜变电压，当连续收到“0”时，则连续下降；

3）正、负斜率相等，且具有记忆功能。上述功能如图 5-29 所示。

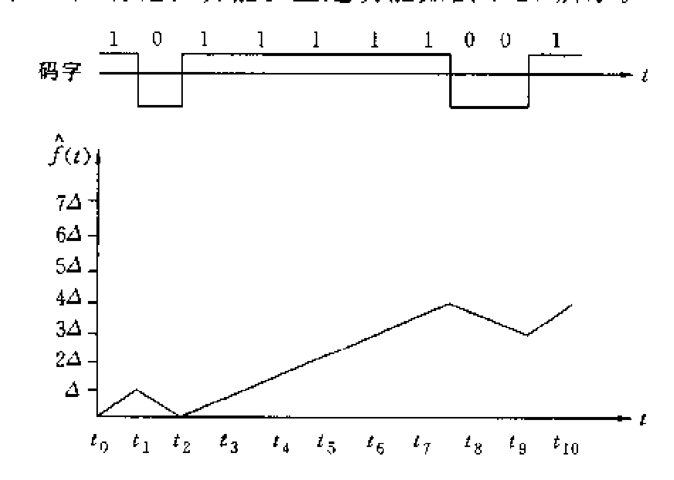


图 5-29 译码原理

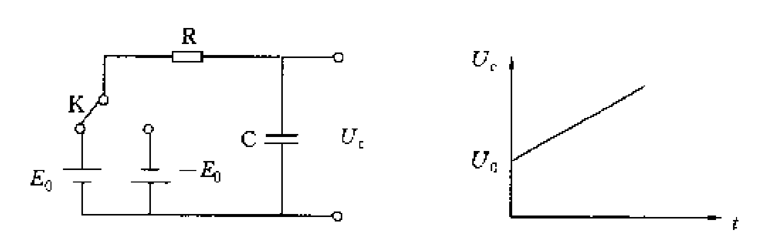


图 5-30 单积分 RC 译码器

最普通的译码器就是一个 RC 积分电路。电路的工作原理如图 5-30 所示。当输入“1”时，开关接 $+E_0$，输入“0”时。开关接 $-E_0$，电容的二端就是译码输出。如果在 $t=0$ 时输入“1”，也就是开关接到 $+E_0$ 上。假定此时电容上已有电压 U_0，则电容器上的电压 U_c 可用式(5-54)求出：

$$U_c = E_0(1 - e^{-\frac{1}{RC}}) + U_0 e^{-\frac{1}{RC}} \tag{5-54}$$

式中，第一项表示 $U_0=0$ 时，E_0 对电容 C 的充电，第二项表示 $E_0=0$ 时 U_0 的放电。当二者都存在时，U_c 是它们的和。因为

$$e^{-\frac{1}{RC}} = 1 - \frac{t}{RC} + \frac{1}{2}\left(\frac{t}{RC}\right)^2 - \frac{1}{2 \times 3}\left(\frac{t}{RC}\right)^3 + \cdots \tag{5-55}$$

这里 t 是一个码元的长度，而 t 远小于 RC，所以式(5-55)可近似为式(5-56)的形式：

$$e^{-\frac{t}{RC}} \approx 1 - \frac{t}{RC} \tag{5-56}$$

这样，在收到"1"时，电容器上的电压为

$$\begin{aligned} U_c &= E_0\left(1 - 1 + \frac{t}{RC}\right) + U_0\left(1 - \frac{t}{RC}\right) \\ &= (E_0 - U_0)\frac{t}{RC} + U_0 \end{aligned} \tag{5-57}$$

式中 U_0 可看作是先前各码元在电容器上建立的电压之代数和。一般情况下，U_0 是远小于 E_0 的，所以，电容器上的电压 U_c 可近似为下式：

$$U_c \approx E_0 \frac{t}{RC} + U_0 \tag{5-58}$$

如果连续收到 n 个"1"，则电容器上的电压可由式(5-59)表示：

$$U_c = E_0 \frac{nt}{RC} + U_0 \tag{5-59}$$

只要 nt 远小于 RC，则电容器上的电压会一直随时间线性增长，保证在收到连"1"码时，每次上升同样一个量化级，上升的斜率就是 $E_0 \frac{t}{RC}$。另外，电容器能够保持电荷，因而具有记忆作用。由式(5-58)知道，收到"1"时电压会上升一个量化阶，当收到"0"时，相当于图 5-30 中开关接到 $-E_0$，此时会使电容上的电压下降一个量化阶，所以，简单的 RC 电路就能实现增量调制编码器的译码。

下面讨论编码器的工作原理。假定"1"码的电压值为 $+E_0$，"0"码的电压值为 $-E_0$。编码原理如图 5-31 所示。

图像信号 $f(t)$ 送入相减器，输出码经本地译码后产生的预测值 $\hat{f}(t)$ 也送至相减器。相减器的输出就是图像信号 $f(t)$ 与其预测值 $\hat{f}(t)$ 之差 $e(t)$，即

$$e(t) = f(t) - \hat{f}(t)$$

误差信号 $e(t)$ 送入脉冲形成器以控制脉冲形成。脉冲形成器一般由放大限幅和双稳判决电路组成。脉冲形成器的输出就是所需要的数码。码率由取样脉冲决定。当取样脉冲到来时刻 $e(t)>0$ 则发"1"，当 $e(t)<0$ 则发"0"。发"0"还是发"1"完全由 $e(t)$ 的极性来控制，与 $e(t)$ 的大小无关。为了提高控制灵敏度，在电路中还加有放大限幅电路。图 5-31 说明了编码过程。在 $t=t_0$ 时，输入一模拟信号 $f(t)$，在此时刻 $f(t_0)>\hat{f}(t_0)$，也就是 $e(t)>0$，则脉冲形成电路输出"1"。从 t_0 开始本地译码器将输出正斜变电压，使 $\hat{f}(t)$ 上升，以便跟踪 $f(t)$。由于 $f(t)$ 变化缓慢，$\hat{f}(t)$ 上升较快，所以在 t_1 时刻 $f(t)-\hat{f}(t)<0$，因此，在第二个时钟脉冲到来时便输出码"0"。以此类推，在 $t_2, t_3, \cdots, t_n$ 等时刻码字的产生原理相同。图 5-31 中分别画出了编出的码流、时钟及误差信号的示意波形。显而易见，$\hat{f}(t)$ 对 $f(t)$ 的跟踪越好，则误差信号 $e(t)$ 越小。这就是 ΔM 编、译码的基本原理。

2. ΔM 编码的基本特性

ΔM 编码性能主要由斜率过载特性、量化噪声以及量化信噪比等性能来衡量。

(1) 斜率过载特性

由 ΔM 的编码原理可知，$\hat{f}(t)$ 应很好地跟踪 $f(t)$，跟踪得越好，误差 $e(t)$ 越小。当 ΔM 编码器出现连"1"或连"0"码时，就说明输入模拟信号 $f(t)$ 有较大的斜率。当判决时钟脉冲的频

率及跳变量化台阶确定后，$f(t)$的最大变化斜率就应满足下式：

$$\left|\frac{\mathrm{d}f(t)}{\mathrm{d}t}\right|_{\max} \leqslant \frac{\Delta}{T_s} \tag{5-60}$$

式中 Δ 代表量化阶，T_s 是取样脉冲周期。

如果输入的是正弦信号，即

$$f(t) = A\sin \omega_c t \tag{5-61}$$

式中 A 是信号 $f(t)$的振幅，ω_c 是正弦波的角频率。当 $t=0$ 时，

$$\left|\frac{\mathrm{d}f(t)}{\mathrm{d}t}\right|_{\max} = A\omega_c \tag{5-62}$$

在这种情况下，不过载条件为

$$A \leqslant \frac{\Delta}{2\pi}\left(\frac{f_s}{f_c}\right) \tag{5-63}$$

式中 f_s 是取样脉冲频率，f_c 是正弦波的频率。一般来说，为了满足不过载条件，ΔM 的取样率要比 PCM 高得多。例如，视频信号的带宽 $f_c=6.5\text{MHz}$，如果采用 PCM 编码 $f_s=2\times f_c=13\text{MHz}$。当每取样值编 8 位码时，码率可达 104Mb。当采用 ΔM 编码时，如果正弦信号峰值 $A=1\text{V}$，量化阶为 $\Delta=0.1\text{V}$，由式(5-63)可求得不过载的 f_s，

$$f_s \geqslant \frac{A}{\Delta}\times 2\pi f_c = \frac{1}{0.1}\times 2\times 3.14\times 6.5 = 408\text{Mb}$$

显然，码率太高了。当然，这只是指避免过载而言。一般情况，不能单靠提高 f_s 的办法来解决过载问题，否则码率太高。解决斜率过载的有效方法是采用自适应增量编码法，即 ADM 编码法。

(2)ΔM 的量化噪声

ΔM 编码法量化噪声的产生如图 5-32 所示。由图可见，在不过载的情况下，量化噪声的幅度不会超过$\pm\Delta$，而且，可认为在$-\Delta\sim+\Delta$范围内量化噪声是以等概率出现的，因此，量化噪声的概率密度可由式(5-64)表示：

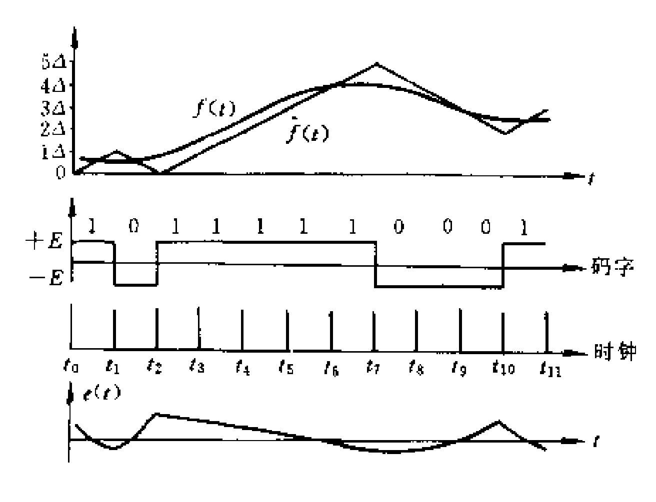


图 5-31　ΔM 编码原理

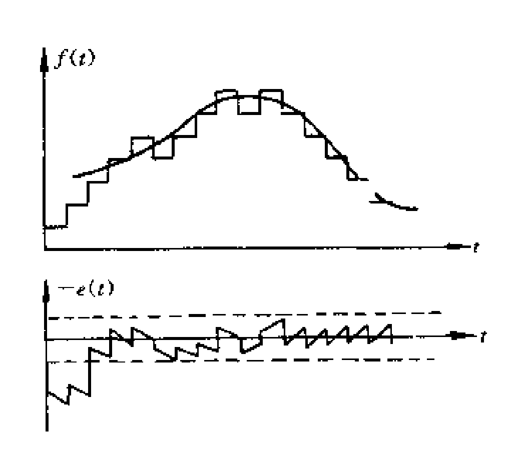


图 5-32　ΔM 编码的量化噪声

$$p(e) = \begin{cases} \dfrac{1}{2\Delta} & -\Delta \leqslant e \leqslant +\Delta \\ 0 & \text{其他} \end{cases} \tag{5-64}$$

量化噪声的功率由式(5-65)表示：

$$N_q=\int_{-\Delta}^{+\Delta}e^2p(e)\mathrm{d}e=\frac{\Delta^2}{3} \tag{5-65}$$

由式(5-65)得到的 N_q 是指在编码器中由比较判决带来的量化噪声功率。它的频谱很宽，并且它的频谱可以近似地认为是均匀分布的，也就是其频谱从低频到高频的分布是一样的。在这样的前提下，可容易地求出它的功率谱密度，即

$$\sigma_N=\frac{\Delta^2}{3f_s} \tag{5-66}$$

在译码时，由于有一个截频为 f_m 的低通滤波器，所以，它将抑制一部分量化噪声。此时，在译码输出端，量化噪声的平均功率由式(5-67)表示。$\overline{N}_q$ 就是 ΔM 编码器的量化噪声，

$$\overline{N}_q=\frac{\Delta^2}{3}\cdot\frac{f_m}{f_s} \tag{5-67}$$

(3) ΔM 的量化信噪比

一般量化噪声的大小并不能完全说明一幅图像质量的好坏。与语音信号编码一样，信号幅度(或功率)与噪声幅度(或功率)的比值才能较全面地说明一幅图像质量受噪声影响的程度。正弦信号的平均功率可由式(5-68)求得：

$$S=\frac{A^2}{2} \tag{5-68}$$

在保证不过载的情况下，A 应满足式(5-69)：

$$A\leqslant\frac{\Delta}{2\pi}\cdot\frac{f_s}{f_c} \tag{5-69}$$

因此，

$$S=\frac{\left(\frac{\Delta}{2\pi}\cdot\frac{f_s}{f_c}\right)^2}{2}=\frac{\Delta^2}{8\pi^2}\cdot\left(\frac{f_s}{f_c}\right)^2 \tag{5-70}$$

由此，可以求得 ΔM 的量化信噪比为

$$\frac{S}{\overline{N}_q}=\frac{3}{8\pi^2}\cdot\frac{f_s^3}{f_c^2\cdot f_m} \tag{5-71}$$

式中 f_s 是取样频率，f_c 是视频信号的最高频率，f_m 是低通滤波器的截止频率。由此可见，在滤波器的截止频率和视频信号的带宽都确定的情况下，ΔM 编码器的量化信噪比与取样频率的三次方成正比。

如果把式(5-71)表示的量化信噪比用分贝来表示可得到下式：

$$\begin{aligned}\left(\frac{S}{\overline{N}_q}\right)_{\mathrm{dB}}&=10\lg 0.04\cdot\frac{f_s^3}{f_c^2\cdot f_m}=10\lg 0.04f_s^3-10\lg f_c^2-10\lg f_m\\&=-14+30\lg f_s-20\lg f_c-10\lg f_m\end{aligned} \tag{5-72}$$

由式(5-72)可以看到，ΔM 的量化信噪比随着 f_s 的增加以每倍频 9dB 的速度增加；随着低通滤波器截止频率的提高以每倍频 3dB 的速度下降；随着视频信号带宽 f_c 的增加以每倍频 6dB 的速度下降。

3. 介绍一种 ADM 系统

解决过载问题的有效方法是利用自适应技术。斜率过载的主要原因在于在常规编码中量化阶是固定的，因此，当信号变化过快时，预测值 $\hat{f}(t)$ 很难跟踪 $f(t)$ 的变化，这时就会产生较

大的误差。自适应增量调制编码方案的基本思想，是根据信号的变化速度相应地改变量化阶梯的大小。这种改变可由系统自动地加以控制。如果比较器连续输出同种极性的误差信号(或者连续出现相同的码字)，则说明输入信号变化剧烈。反之，如果比较器输出极性交替变化的误差信号，则说明信号变化缓慢。由此可见，可以用误差信号的极性去控制量化阶梯的长度。变化剧烈的信号出现时，加大阶梯步长，以便更好地跟踪输入信号的变化，避免斜率过载。反之，如果出现交变极性的误差信号(或者说交替出现"0"、"1"码型)，说明信号变化缓慢，因此，可以利用误差信号极性或者码型的变化去控制量化阶梯的长度。变化剧烈则增加阶梯长度，以便更好地跟踪输入信号的变化，避免斜率过载。变化缓慢则减小步长，以便减小量化噪声。这样，就可以用不太复杂的技术既减少码率，又可以得到较好质量的编码图像。

目前，所应用的自适应增量调制器大致有三种系统。第一种是用前三个输出电平的极性使阶梯步长与信号变化相匹配。根据前三个输出电平极性可有八种可能的组合，与其相匹配使用±1，±2，±4 几个阶梯尺寸。第二种方法叫"桑"(song)增量调制系统。它的基本原理是使用前一样值的阶梯步长，形成当前样值的阶梯步长。第三种是杰伊亚特(Jayant)提出的带有 1 比特存储器的自适应增量调制器。它有一个存入前一输出比特极性的存储器，样值的阶梯步长是变化的。除此之外，还有自适应改变取样速率的 ADM 系统。

图 5-33 是一种自适应增量调制编、译码器原理框图。本方案的编码器和译码器分别如图(a)和(b)所示。实际上就是在基本 DM 方案上加上一个自适应环路构成的。这个自适应环路由三个 D 触发器构成一个移位寄存器以及一个逻辑器组成。它是利用三个输出电平的极性使步长与信号变化相匹配。这三个电平有八种可能的组合，使用 1，2，2，3 的阶梯步长比自适应地改变量化阶。其中最小步长的优值由式(5-73)表示：

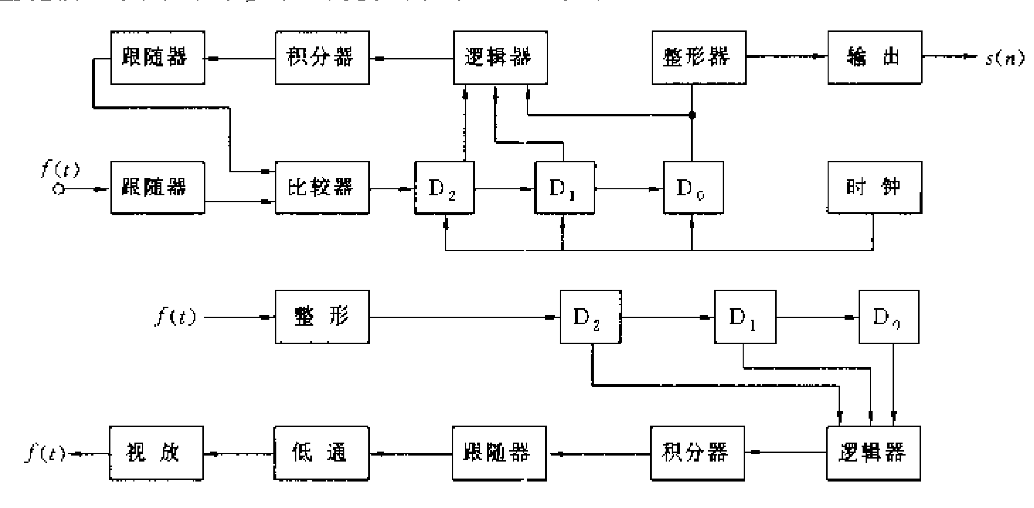


图 5-33 一种自适应增量调制编、译码原理框图

$$\Delta_{min} = 0.26\sigma_1 \frac{2f_c}{f_s}\ln\left(\frac{f_s}{f_c}\right) \tag{5-73}$$

式中 σ_1 是输入信号的最小标准偏差，f_c 是视频信号的带宽，f_s 是取样钟频。

这个方案的关键部分是自适应逻辑控制部分。这部分主要包括移位寄存器和逻辑控制电路。移位寄存器的电路如图 5-34 所示。它是由三个 D 触发器 D_0，D_1，D_2 组成。图中将触发器 D_2 的 Q 端定为 A_2，D_1 的 Q 端定为 A_1，D_0 是 Q 端定为 A_0。这样，由 $A_2A_1A_0$ 的码元组合信息就可以指出输入信号是处于急剧变化状态还是处于平缓变化状态。例如，当出现连码状态，即

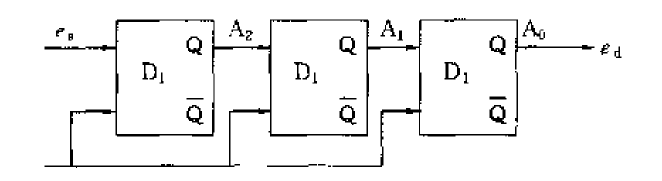


图 5-34　移位寄存器电路图

$A_2A_1A_0$ 为 111 或 000 时，就说明输入信号处于急剧变化状态，此时，阶梯步长应加大，以保证良好的跟踪。如果码元组合为 101 或 010 时，则说明信号处于平缓变化状态，阶梯步长应减小，以保证小的量化噪声。由于 $A_2A_1A_0$ 共有 8 种组合，每种组合为一个状态，共有 8 种状态。组合与状态的对应关系如表 5-13 所示。

表 5-13　状态真值表

状　态	3	2′	2	1	−1	−2	−2′	−3
A_2,A_1,A_0	111	110	100	101	010	011	001	000

上述逻辑的状态转移图如图 5-35 所示。从状态转移表中的对应关系不难理解状态转移图。逻辑设计必须遵循如下原则：

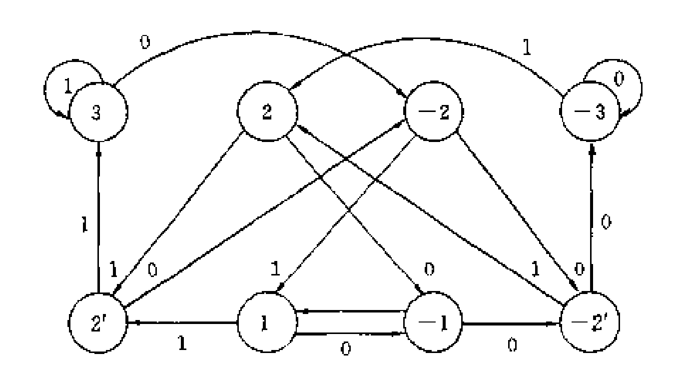


图 5-35　状态转移图

1）对于同一码元序列状态转移的通路应该一致；

2）如果码元序列出现 0、1 交替码，即 1010…，最后应稳定在最小阶梯步长状态；

3）当模拟输入任意大小的阶跃信号时，应能尽量匀称地跟踪，然后稳定在最小阶梯步长状态，以免产生过冲；

4）信噪比性能良好。

例如，当状态处在状态转移图的②处时，此时码元序列为 1101010001，与此相对应的通路只有一条，即如图 5-36 所示。另外，当码元序列为 101010…时，无论从哪里开始都会稳定在+1和−1 的状态上。

以上便是 ADM 的自适应逻辑设计。逻辑控制器和积分器电路原理如图 5-37 所示。译码器逻辑关系与编码器相同。这一全电视信号编码器最小阶梯步长为 0.07V，步长比例关系为 1，2，2，3，取样频率 f_s=32MHz，视频输入信号幅度为 1V，视频带宽 f_c=4.5MHz，信噪比可达

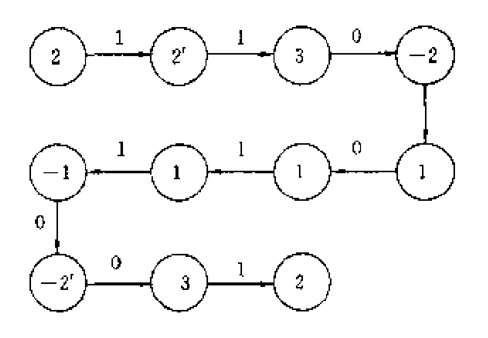


图 5-36　对应于码元序列的唯一通路

29.7dB。当最小步长采用 0.035V 时，信噪比可做到 35dB。

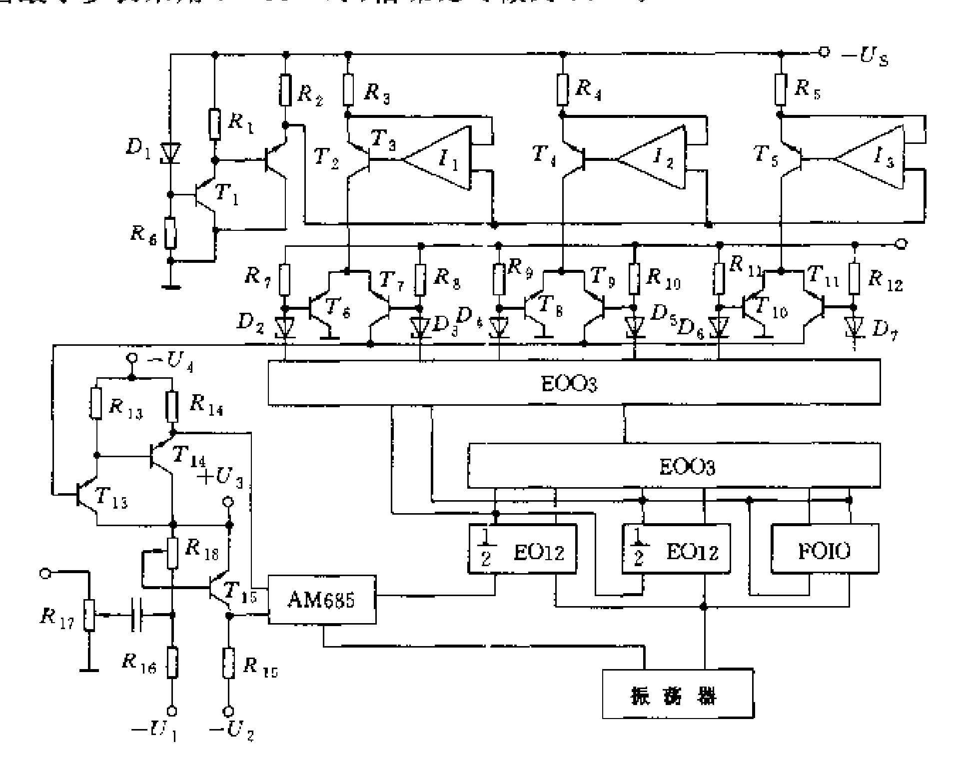


图 5-37　ADM 逻辑控制器和积分器电路原理图

5.5.3　DPCM 编码

预测编码的另一种有用的形式是 DPCM 编码(Differential Pulse Code Modulation)，这实际上是 ΔM 和 PCM 两种技术相结合的编码方法。

1. DPCM 编码的基本原理

在卡特勒的专利中提出利用积分器根据一行上前样本值预测现样本值，并且把现样本值与其估计值的差值进行量化和编码。这就是 DPCM 的基本设计思想。DPCM 编码的基本原理如图 5-38 所示。图中(a)是编码器原理框图。它由取样器、比较器、量化器、预测器、编码器五个部分组成。输入信号经采样后将样值 $f(t)$ 送入比较器，使得 $f(t)$ 与预测值 $\hat{f}(t)$ 相减得出误差信号，即 $e(t)=f(t)-\hat{f}(t)$。然后，将 $e(t)$ 送入量化器量化为 M 个电平之一($M=2^N$)，量化后的样值再送入 PCM 编码器中编码，以便传输。另外一路是将 $e(t)$ 送入相加器，在这里 $e(t)$ 与

$\hat{f}(t)$相加后再送入预测器，以便预测下一个样值。译码器的原理框图如图(b)所示。译码器收到码字后首先经 PCM 译码，得到 $e(t)$后再送入相加器与预测值 $\hat{f}(t)$相加得到 $f(t)$。另外，$f(t)$又送到预测器以便预测下一个样值。

由上面的原理可知，DPCM 实际上是综合了 ΔM 和 PCM 两种编码技术的一种编码方法，ΔM 是一位二进制码的差分脉码调制，也就是用 1bit 码来表示增量值，而 DPCM 是 N 位二进码来表示 $e(t)$值的编码法，因此 DPCM 编码原理又可简化为图 5-39 所示的形式。

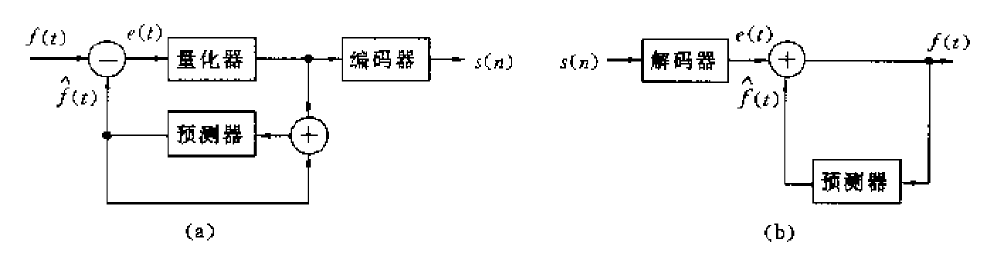


图 5-38 DPCM 编、译码原理框图

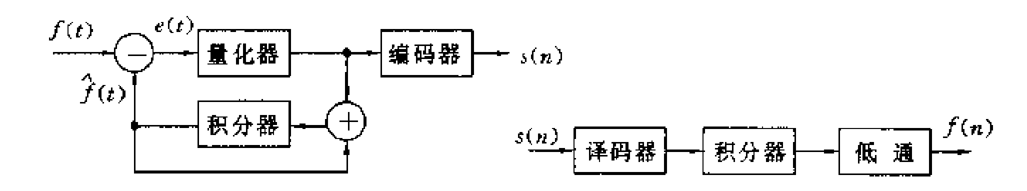


图 5-39 DPCM 原理框图

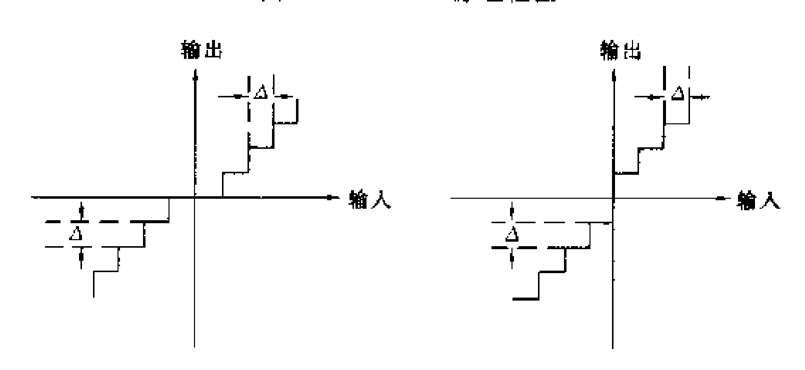


图 5-40 两种均匀量化的输入、输出特性

2. DPCM 编码的量化信噪比

DPCM 编码器中的量化器与 PCM 中的量化器具有相同的工作原理。如图 5-40 所示，量化器的特性有(a)，(b)两种。这两种特性在小信号输入情况下有比较明显的差别，对于(a)特性来说，当输入值在 0～Δ 之间时，量化器没有输出。但是，对于(b)特性来说则有输出。在输入信号幅度大时是没有区别的。图中的一个阶梯 Δ 就是一个量化阶。由于在整个输入信号幅度范围内量化阶 Δ 是一个常数，所以称为均匀量化。

由于 DPCM 编码仍然是对误差信号编码，所以其不过载条件仍然要满足下式，即

$$A \leqslant \frac{\Delta}{2\pi} \cdot \frac{f_s}{f_c}$$

在临界状态下，

$$A = \frac{\Delta f_s}{2\pi f_c} \tag{5-74}$$

系统最大信号功率输出为

$$S=\frac{\Delta^2 f_s^2}{8\pi^2 f_c^2} \tag{5-75}$$

但是，由于误差的范围是在$(+\Delta,-\Delta)$之间，在 DPCM 系统中，误差又被量化为 M 个电平，则

$$\Delta=\left(\frac{M-1}{2}\right)\cdot\delta \tag{5-76}$$

式中 δ 是 DPCM 量化阶。将式(5-76)代入式(5-75)，则有

$$S=\frac{\left(\frac{M-1}{2}\right)^2\delta^2 f_s^2}{8\pi^2 f_c^2}=\frac{(M-1)^2\delta^2 f_s^2}{32\pi^2 f_c^2} \tag{5-77}$$

这是在临界过载条件下的最大输出功率公式。其中 M 是量化级数，δ 是 DPCM 量化阶，f_s 是取样频率，f_c 是视频信号频带宽度。

在 DPCM 中，由于系统的量化误差不再在 $\pm\Delta$ 范围内，而是在$\left(-\frac{\delta}{2},+\frac{\delta}{2}\right)$范围内，其中 $\delta=\frac{2\Delta}{M-1}$。

由于对 $e(t)$ 的编码是 PCM 编码，所以其量化噪声应符合 PCM 编码量化噪声规律，即

$$N_q'=\frac{\delta^2}{12} \tag{5-78}$$

如果 DPCM 系统输出数字信号的码元速率为 Nf_s，同时，可认为噪声频谱均匀地分布于频带宽度为 Nf_s 的范围内，这时可求得量化噪声功率谱密度为

$$p(f)=\frac{\delta^2}{12Nf_s} \tag{5-79}$$

式中 N 是编码比特数，f_s 为取样频率，δ 为量化阶，在译码时，考虑到低通滤波器的作用，则噪声功率为

$$N_q=p(f)\cdot f(m)=\frac{\delta^2}{12Nf_s}\cdot f(m) \tag{5-80}$$

因此，可求得 DPCM 编码的量化信噪比为

$$\left(\frac{S}{N_q}\right)=\frac{\frac{(M-1)^2\delta^2 f_s^2}{32\pi^2 f_c^2}}{\frac{\delta^2}{12Nf_s}\cdot f_m}=\frac{3N(M-1)^2 f_s^3}{8\pi^2 f_c^2\cdot f_m} \tag{5-81}$$

式中 S 代表信号功率，N_q 代表噪声功率，f_m 是低通滤波器的截止频率，N 是编码的比特数，其他符号的意义同前。式(5-81)便是 DPCM 编码的信噪比性能。与 ΔM 编码的性能作一下比较。如前所述，ΔM 的量化信噪比为

$$\left(\frac{S}{N_q}\right)=\frac{3}{8\pi^2}\cdot\frac{f_s^3}{f_c^2\cdot f_m}$$

而 DPCM 的量化信噪比为

$$\left(\frac{S}{N_q}\right)=\frac{3N(M-1)^2}{8\pi^2}\cdot\frac{f_s^3}{f_c^2\cdot f_m}$$

显然在 f_s 相同的情况下，

$$\frac{3N(M-1)^2}{8\pi^2}\gg\frac{3}{8\pi^2}$$

这说明 DPCM 的性能远优于 ΔM。在 $N=1$，$M=2$ 的情况下，DPCM 就变成 ΔM 编码法了，其量化信噪比自然也就等于 ΔM 的量化信噪比。与 ΔM 编码方法一样，在 DPCM 编码中为了适应非平稳信号的特性，常采用可变量化器。这也是一种自适应方式。目前，DPCM 编码的自适应方案设计中，有两种途径可循，一是采用可变参数预测器，其参数随着信号的变化而变化，这样可产生一个平稳的差分信号；其次是采用固定预测器，而采用一个可变的量化器去适应所得到的非平稳的差分信号。除此之外，使用可变的取样速率，而预测器和量化器都是固定的也应属于自适应的范畴。据文献报导，采用自适应技术较采用非自适应技术编码可使图像信噪比提高 10dB。

另外，曾经提出一种自适应改变编码方式的技术，如双模式预测编码。这种方法联合使用 ΔM 和 DPCM 编码技术。当视频信号变化缓慢时采用 ΔM 编码法，而变化较快时采用 DPCM 编码法。据称，对于单色、单帧图像，每像素 1.5bit 就会有较好的图像质量。当然，设备相应的也就比较复杂了。

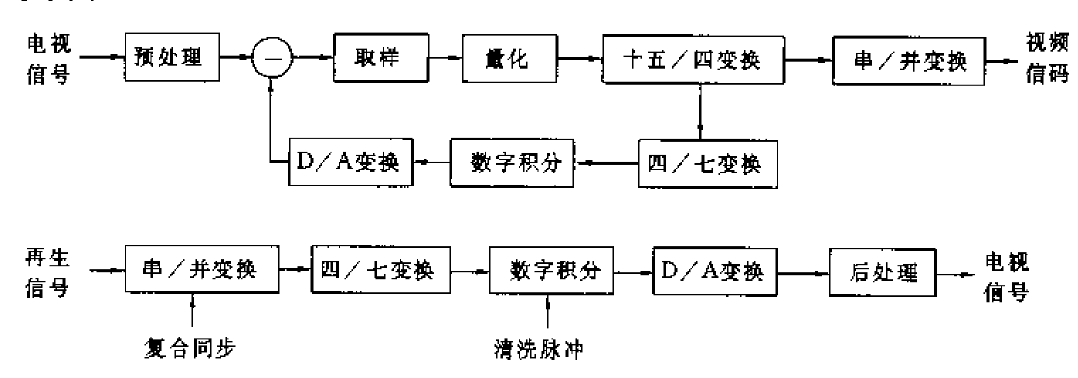


图 5-41　一种用于可视电话的 DPCM 编译码原理框图

3. 一种用于可视电话的 DPCM 编码装置

可视电话的视频带宽一般较电视频带宽度为窄，大约在 1MHz 左右。图 5-41 是一种可视电话 DPCM 视频信号编、译码原理框图，图中(a)为编码器，(b)为译码器。

编码器由预处理电路、相减器、采样器、量化器、编码器、四/七变换、数字积分器、D/A 变换器及并串变换等部分组成。在编码过程中，首先将全电视信号送人预处理器，预处理的目的在于滤除高频信号，限制带宽以及削去同步头，并且将信号放大到所需要的幅度。经预处理的视频信号和预测器送来的预测信号在相减器中相减，误差信号送往编码器。编码器由采样器、量化器及编码器组成，采样时钟由采样脉冲发生器提供。量化器将误差信号经时间离散化后的样值分为 16 层不同的电平值，这量化后的样值送人编码器编成 4 位并行码。并行码分两路送出，一路送人并-串变换电路，以便把四位并行码变成便于传输的串行码。另一路送人预测器。预测器由四/七变换电路、数字积分器及 D/A 变换器组成。四/七变换电路把四位并行码变成七位码，以便于数字运算。数字积分器是并行进位的加法器。在这里，第 n 个误差值与第 $n-1$ 个预测值相加以便产生新的预测值。相加后的数字信号送人 D/A 变换器变为模拟信号。这个模拟信号送人相减器与送来的视频信号相减以便产生新的差值。

收端译码器由串-并变换器、四/七转换器、数字积分器、D/A 变换器及后处理器组成。串-并变换器把收到的串行码变为并行码，然后把并行码送人四/七变换器，变换后送入数字积分器，以便产生数字式恢复值，这个数字信号经 D/A 变换恢复为模拟视频信号。实际上译码中由四/七变换、数字积分器、D/A 转换器组成的预测器与编码器中的预测器基本一样。为了防止

信道误码在数字积分器中的积累，可加入清洗脉冲，以便在行同步期间清除寄存器。后处理器主要完成滤波及叠加同步头的作用。这样就恢复了完整的全电视信号送入电视机显示。

5.6 变换编码

图像编码中另一类有效的方法是变换编码。变换编码的通用模型如图 5-42 所示。



图 5-42 图像变换编码模型

变换编码主要由映射变换、量化及编码几部分操作组成。映射变换是把图像中的各个像素从一种空间变换到另一种空间，然后针对变换后的信号再进行量化与编码操作。在接收端，首先对接收到的信号进行译码，然后再进行反变换以恢复原图像。映射变换的关键在于能够产生一系列更加有效的系数，对这些系数进行编码所需的总比特数比对原始图像进行编码所需要的总比特数要少得多，因此，使数据率得以压缩。

映射变换的方法很多。广义地讲，前面讨论的预测编码法也可称为是预测变换。它是将信号样值的绝对值映射为相对样元的差值，只是根据实用技术上的习惯，没有把它归入变换编码的范畴罢了。图像变换编码基本可分为两大类，一类是某些特殊的映射变换编码法，另一类就是函数变换编码法。

5.6.1 几种特殊的映射变换编码法

特殊映射变换编码法包括诸如行程编码，轮廓编码等一些变换编码方法。它们特别适用于所谓二值图像的编码。这类图像包括业务信件、公文、气象图、工程图、地图、指纹卡片及新闻报纸等。当然在编码技术上同样可分为精确编码和近似编码两类。精确编码可以不引入任何畸变，在接收端可以从编码比特流中精确恢复出原始图像。近似编码会引入一些畸变，但是，这种方法却可以在保证可用性的前提下获得较高的压缩比。下面通过几种具体的编码方法说明这种变换编码法的基本概念。

1. 一维行程编码

一维行程编码的概念如图 5-43 所示。

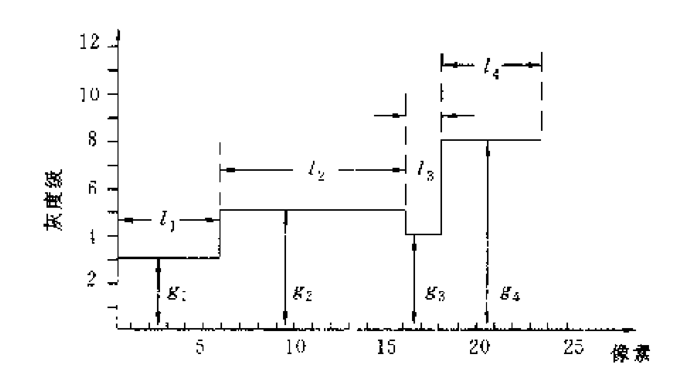


图 5-43 一维行程编码

假如沿着某一扫描行的像素为 $x_1,x_2,\cdots,x_N$，它们所具有的灰度值可能为 g_1,g_2,g_3,g_4。在编码之前，可以首先把这些像素映射为成对序列(g_1,l_1)，(g_2,l_2)，(g_3,l_3)和(g_4,l_4)。其中 g_i 表示某一灰度值，l_i 表示第 i 次运行的行程。也可以说是连续取值为 g_i 灰度值的像素的个数。经过这样映射变换后就可以对(g_i,l_i)编码，而不必对像素直接编码。由于有些图像如前面提到的二值图像，连续取同一灰度级的像素很多，对映射后的序列进行编码会大大压缩比特率。例如，图 5-43 所示的例子可映射成表 5-14 所示的序列对。在这个例子中有 8 级灰度，24 个像素。如果对 x_i 编码，总的比特数至少要 $24\times3=72$bit。如果对表 5-14 的序列对编码，灰度值用 3 位码，行程长度用 4 位码，每对参数用 7 位码，共 4 对，总比特数只要 28bit 就够了，可见压缩率是很可观的。

表 5-14 序列对

i	g_i	l_i
1	3	6
2	5	10
3	4	2
4	8	6

行程编码可分为行程终点编码和行程长度编码。如果行程终点的位置由扫描行的开始点算起，并且由到达行程终点的像素计数来确定，就称为行程终点编码。如果行程终点位置由这一终点与前一终点的相对距离确定，就称为行程长度编码。

对于二值图像来说采用行程长度编码，甚至不需要传送灰度信息。假定某一扫描线含有 3 个白色像素，其后是 2 个黑色像素，接着又是 10 个白色像素。这样，在行程长度编码中，只传送行程长度 3、2 和 10 就可以了。每个行程长度告诉沿扫描线的下一个边界点的相对位置。

行程长度编码的比特率可作如下估计。行程长度编码的消息集合含有行程长度。1,2,…,N。这里 N 是一条扫描线中的像素数目。如果测得行程长度的概率为 $P_1,P_2,\cdots,P_N$，且用统计编码法，则每个行程的比特率 B 应满足式(5-82)：

$$H \leqslant B \leqslant H + 1 \tag{5-82}$$

其中 H 就是行程长度的熵：

$$H = -\sum_{i=1}^{N} P_i \log_2 P_i \tag{5-83}$$

如果令 v 为平均行程长度，

$$v = \sum_{i=1}^{N} i P_i \tag{5-84}$$

则每个像素的比特率应满足式(5-85)：

$$\frac{H}{v} \leqslant b \leqslant \frac{H}{v} + \frac{1}{v} \tag{5-85}$$

这就是采用行程长度编码可能得到的比特率的估计。如果把黑白行程长度分开编码，有可能进一步降低行程长度编码的比特率。可以把消息集合分成两个子集，一个含有白色行程长度，另一个含有黑色行程长度。若对每个子集用统计编码法，则每个像素的比特率满足式(5-86)：

$$h_{WB} \leqslant b \leqslant h_{WB} + \left(\frac{P_W}{v_W} + \frac{P_B}{v_B}\right) \tag{5-86}$$

其中

$$h_{WB} = P_W \frac{H_W}{v_W} + P_B \frac{H_B}{v_B} \tag{5-87}$$

式中 P_W 为白色像素概率，P_B 为黑色像素概率，并且 $P_W+P_B=1$。H_W 为白色行程长度的熵：

$$H_W = -\sum_{i=1}^{N} P_{W_i} \log_2 P_{W_i} \tag{5-88}$$

H_B 为黑色行程的熵：

$$H_B = -\sum_{i=1}^{N} P_{B_i} \log_2 P_{B_i} \tag{5-89}$$

式中 P_{W_i}是白色行程的概率，P_{B_i}为黑色行程的概率。v_W为平均白色行程长度：

$$v_W = \sum_{i=1}^{N} i P_{W_i} \tag{5-90}$$

v_B 为平均黑色行程长度：

$$v_B = -\sum_{i=1}^{N} i P_{B_i} \tag{5-91}$$

如果对黑白色行程长度分别用统计编码法，有可能使 $h_{WB} \leqslant \frac{H}{v}$。

与其他统计编码的难点一样，行程长度的概率是很难测量的。如果采用一阶马尔可夫模型，那么仅测量平均行程长度就能较好地估计出比特率。作为一阶马尔可夫信源的图像特征是转移概率，即 $P(B/W)=q_0$，$P(W/W)=1-q_0$，$P(W/B)=q_1$，$P(B/B)=1-q_1$。这里$P(B/W)$表示在给定的第 k 个像素是白色时，第 $k+1$ 个像素是黑色的概率。在这种条件下，可求得黑白像素的先验概率：

$$\begin{aligned} P_W &= \frac{q_1}{(q_0+q_1)} \\ P_B &- \frac{q_0}{(q_0+q_1)} \end{aligned} \tag{5-92}$$

黑白色行程长度概率为

$$\begin{aligned} P_{W_i} &= q_0(1-q_0)^{i-1} \\ P_{B_i} &= q_1(1-q_1)^{i-1} \end{aligned} \tag{5-93}$$

黑白色行程的平均长度为

$$\begin{aligned} v_W &= \frac{1}{q_0} \\ v_B &= \frac{1}{q_1} \end{aligned} \tag{5-94}$$

这样就可估计出一阶马尔可夫模型下的每个像素的比特率。根据黄的报告，对气象图和印刷品进行行程长度编码，算得每像素的熵最大值为 0.37bit，最小值为 0.28bit。

2. 二维行程编码

二维行程编码也叫预测微分量化器(Predictive Differential Quantizer)，简称 PDQ。其基本算法如图 5-44 所示。

PDQ 的基本算法是将图像元素阵列变换为整数对 Δ'和 Δ''的序列。这里 Δ'和 Δ''的意义如图 5-44 所示。Δ'是相邻扫描行上行程的开始点之间的差。图中 Δ'是 A 点和 B 点的差。Δ''是这相邻行行程的差。对应于 A 起始点的行程为 l_1，对应于 B 起点的行程为 l_2，因此，$\Delta''=l_2-l_1$。另

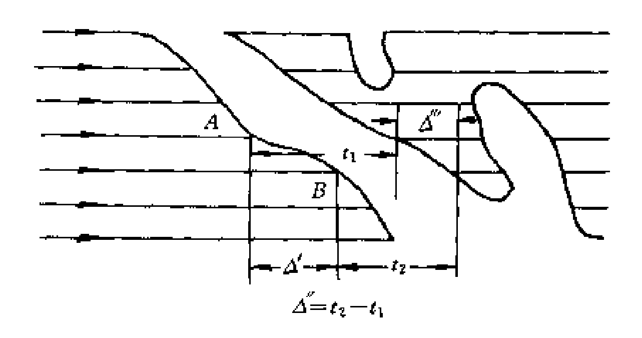


图 5-44 PDQ 算法说明

外，对于图中的暗面积还要有一个“开始”和“消失”的标记。这样就把一幅图像的像素阵列按相继扫描行变换为 Δ′、Δ″、开始、消失 4 个参量的序列，然后便可对这四个参量来编码。PDQ 法利用了扫描线间的相关性，因此，它有更大的压缩潜力。另外一种方法叫做双重增量编码(Double Delta Coding)，简称 DDC。它是对 Δ′和 Δ‴进行编码而不是对 Δ′和 Δ″编码。Δ‴是前一扫描行的暗区后边界与相继扫描行暗区后边界的差。实验证明，这种方法的压缩比较 PDQ 法更大。

当在图像中有少数大的暗区时二维行程编码更有效，对于有许多小暗区的图像来说，一维行程编码更有效。

3. 等值线编码

一幅数字图像可看成是含有两个变量的二元函数。这两个变量就是像素在空间的位置，而函数值就是该像素的灰度值。对数字图像来说灰度值是一个离散的有限的数量。这样，可以把函数想像为许多台阶，台阶的高度就是灰度级。暗的灰度对应着低的台阶，亮的灰度对应着高的台阶。具有同一灰度级的像素就构成了一个“平台”。这样，对所有的平台的高度、位置和形状的了解，也就等于对图像的了解。这就是等值线编码的基本原理。等值线编码的关键是确定等值线，确定等值线的三点要素为：确定等值线的灰度级；确定一个起始点(IP)；为了跟踪等值线的外部边缘而应遵循的移动方向，也就是确定指向符序列。这样，等值线编码法就有两种算法，一个是确定起始点的 IP 算法、一个是跟踪等值线的 T 算法。其中，IP 算法可以给所有的等值线定位，但一条等值线不能被定位两次。T 算法可以对与起始点的灰度值相同的元素集合的外边界跟踪，而且最终总是返回起始点。

(1) T 算法

T 算法是决定跟踪方向的算法。它规定永远向左转，也就是当从一个元素移出而进入另一个元素时，相对于进入方向总是向最左看。这一规则使得自起始点开始，所有落在等值线外边而又邻近等值线的任何一个元素，不会与等值线上的元素有相同的灰度级。其基本方法如图 5-45 所示。向最左看的规则称为 LML 规则。

LML 规则规定，相对于进入方向在某个像素处观察与其相邻的左边的像素，如果该像素与所在像素的灰度值相同，就移向这个像素。如果不相同就再观察上面的像素。如果上面的像素与所在像素的灰度值相同，就移向上面的像素。如果仍然不同，就向右看，如果右边的像素灰度值与所在像素相同，就移向右边的像素。如果仍然不相同，就观察下边的相邻像素，下边的像素如果与所在像素的灰度值相同就移向下边的像素。否则，就没有一个像素与所在像素的灰度值相同，那么这个点就是一个孤立点。以上方法就是 LML 规则的具体运算方法。

T 算法除了有上述 LML 规则外，还要将 4 个指示符之一分配给二维阵列中的每个像素。这 4 个指示符分别是 D、A、R 和 I。当二维数据阵列存入存储器时，将指示符 I 分配给每个像

素。当从一个像素移到另一个像素时，将按着 IA 规则对每个像素以 D、A 或 R 指示符中的一个来代替原来的指示符。IA 规则如图 5-46 所示。

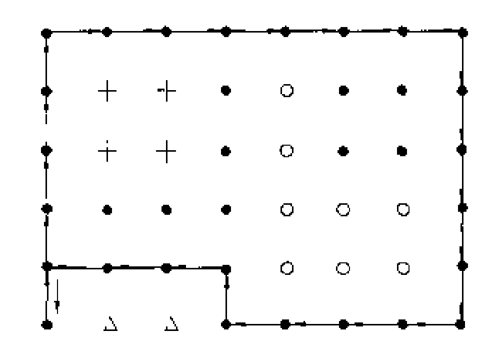
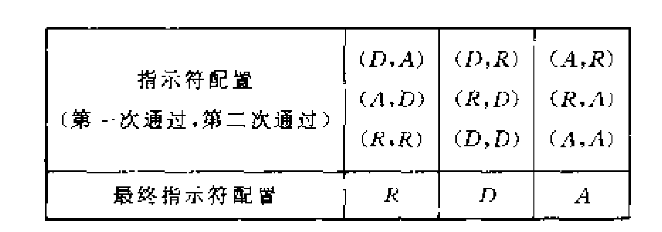


图 5-45　具有四级灰度阵列的 1 号等值线

移入元素方向 \ 移出元素方向	↑或→	↓或←
↑或←	A	R
↓或→	R	D

图 5-46　对移入移出像素各种方向可能组合所设置的指示符

分配给每一等值线像素的指示符取决于移入和移出该像素的方向。有某些像素被通过两次，当第二次通过这个像素时，首先根据图 5-46 来决定这次通过的指示符，然后再根据图 5-47 决定最后分配给该像素的指示符。这里，起始点是例外的，它的指示符总保留为 I。利用 IA 规则可定出图 5-45 所示等值线上像素的指示符。其结果如图 5-48 所示。



指示符配置（第一次通过，第二次通过）	(D,A) (A,D) (R,R)	(D,R) (R,D) (D,D)	(A,R) (R,A) (A,A)
最终指示符配置	R	D	A

图 5-47　对第一次通过和第二次通过所决定的每对指示符的最终配置

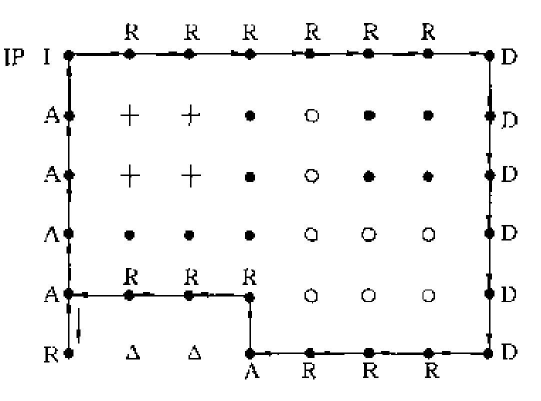


图 5-48　指示符的分配

通过上述规则，在等值线上的所有像素都被赋予一个指示符，而不在等值线上的像素以及起始点都将保留指示符 I。以上便是 T 算法的内容。

(2) IP 算法

IP 算法是寻找起始点的算法。为了寻找起始点，该算法规定从左上角开始(即从第一行第一列开始)，由左至右直至第一行终点为止，此后，从左至右扫描第二行，第三行，直至扫完整个阵列。

寻找 IP 点时还要作一个比较点表(CPL)。此表的构成规则如下：在开始时此表为空的，当扫描一行时，分别检验每一个像素的指示符，如果指示符是 A 就把该像素的灰度值加到此表的后面；如果指示符是 D 就删去最后项目的值；如果指示符是 I 或 R 则保持此表不变。对于每一行来说，删去的数目等于加上的数目，故在每一行终点比较点表是空的。

IP 规则规定，某像素是 IP，它就必须满足两条要求：指示符是 I；灰度值不等于 CPL 最后项目的值。

IP 算法总是从像素阵列的第一行，第一列开始，这一点总是一个 IP，选定 IP 后，用 T 算法

跟踪，直至返回到 IP：此后，再用 IP 算法寻找第二个 IP，然后再用 T 算法跟踪，当返回到 IP 点后就可得到第二条等值线，其余以此类推。与此同时，给每个像素设置一个指示符。这样得到的每条等值线包括 4 个参数：此等值线的灰度级；IP 所在的行号；IP 的列号；指向符序列。

当将一幅图像映射变换为等值线后就可进行编码。一种常用的编码法是对等值线灰度值、IP 的行号、列号三个参量用自然二进码编码，对指向符采用弗利曼链码(Freeman's Chain Code)。此码规则示于图 5-49，弗利曼链规定，向上移动为 00，向右移动为 01，向下移动为 10，向左移动为 11。

对于译码来说，至关重要的是要知道旧等值线已终止，新等值线开始的时刻。因为所有的等值线必定返回和终止于起始点，所以，只要注意左右方向以及上下方向指向符的累加数目就行了。如果左右方向指向符的累加数目为 0，上下方向指向符的累加数目也是 0，那么下面的数据一定属于新的等值线。

等值线编码方法实例可见于表 5-15。这个表是针对图 5-45 的结果。图 5-50 为图 5-45 的 4 个 IP 点及其等值线。

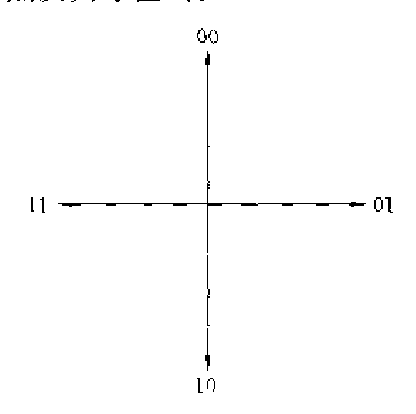


图 5-49　弗利曼链

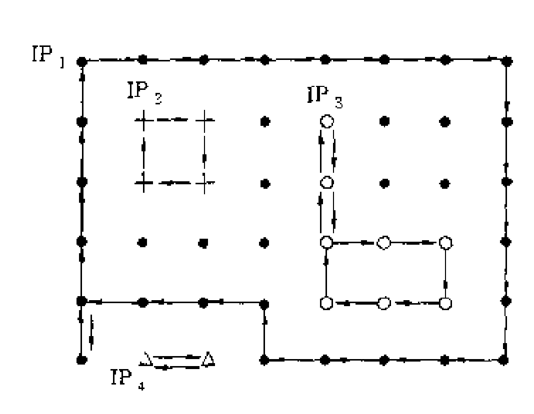


图 5-50　4 个 IP 点及其等值线

表 5-15　等值线编码

等值线	码 字	灰度级	码 字	行或列	码 字	移动方向	码 字
1	0 0	•	0 0	1	0 0 0	↑	0 0
2	0 1	×	0 1	2	0 0 1	→	0 1
3	1 0	o	1 0	3	0 1 0	↓	1 0
4	1 1	Δ	1 1	4	0 1 1	←	1 1
				5	1 0 0		
				6	1 0 1		
				7	1 1 0		
				8	1 1 1		

等值线号　　{00,01,10,11}

IP 值　　{00,01,10,11}

IP 行　　{000,001,001,101}

IP 列　　{000,001,100,011}

IP 后第一个元素的方向　　{01,01,10,01}

IP 后第二个元素的方向　　{01,10,10,11}

⋮

（后边还有 44bit）

一般说来，等值线编码的比特平均数正比于图像的复杂性。

5.6.2 正交变换编码

变换编码中另一类方法是正交变换编码法（或称函数变换编码法）。这种方法的基本原理是通过正交函数变换把图像从空间域转换为能量比较集中的变换域。然后对变换系数进行编码，从而达到缩减比特率的目的。

1．正交变换编码的基本概念

正交变换编码的基本原理框图如图 5-51 所示。编码器由预处理、正交变换、量化与编码几部分组成，译码器由译码、反变换及后处理组成。在编码操作中，模拟图像信号首先送入预处理器，将模拟信号变为数字信号。然后把数字信号分块进行正交变换，通过正交变换就使空间域信号变换到变换域。然后对变换系数进行量化和编码。在信道中传输或在存储器中存储的是这些变换系数的码字。这就是编码端的处理过程。在译码端，首先将收到的码字进行译码，然后进行反变换以使变换系数恢复为空间域样值，最后经过处理使数字信号变为模拟信号以供显示。

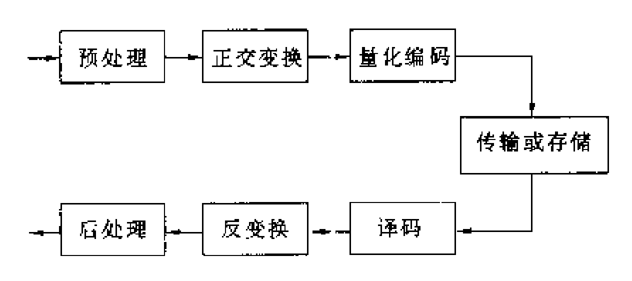


图 5-51　正交变换编码原理框图

正交变换编码之所以能够压缩数据率，主要是它具有如下一些性质：

1）正交变换具有熵保持性质。这说明通过正交变换并不丢失信息，因此，可以用传输变换系数来达到传送信息的目的。

2）正交变换有能量保持性质。这就是第 3 章提到的各种正交变换的帕斯维尔能量保持性质。它的意义在于：只有当有限离散空间域能量全部转移到某个有限离散变换域后，有限个空间取样才能完全由有限个变换系数对于基础矢量加权来恢复。

3）能量重新分配与集中。这个性质使我们有可能采用熵压缩法来压缩数据。也就是在质量允许的情况下，可舍弃一些能量较小的系数，或者对能量大的谱点分配较多的比特，对能量较小的谱点分配较少的比特，从而使数据率有较大的压缩。

4）去相关特性。正交变换可以使高度相关的空间样值变为相关性很弱的变换系数。换句话说，正交变换有可能使相关的空间域转变为不相关的变换域。这样就使存在于相关性之中的多余度得以去除。

综上所述，由于正交变换的结果，相关图像的空间域可能变为能量保持、集中且为不相关的变换域。如果用变换系数来代替空间样值编码传送时，只需对变换系数中能量比较集中的部分加以编码，这样就能使数字图像传输或存储时所需的码率得到压缩。

2. 变换编码的数学模型分析

由正交变换编码的基本概念不难看出，编码过程主要是在变换域上进行。在这个基础上可以建立以下变换编码的数学模型。

设一图像信源为一向量$[X]^T$，

$$[X]^T = [X_0, X_1, X_2, \cdots, X_{N-1}] \tag{5-95}$$

变换后输出一向量$[Y]^T$，

$$[Y]^T = [Y_0, Y_1, Y_2, \cdots, Y_{N-1}] \tag{5-96}$$

取正交变换为$[T]$，那么$[X]$与$[Y]$之间的关系为

$$[Y] = [T][X] \tag{5-97}$$

由于$[T]$是正交矩阵，所以

$$[T][T]^T = [I] = [T]^{-1}[T] \tag{5-98}$$

这里$[I]$为单位矩阵，$[T]^T$是$[T]$的转置，$[T]^{-1}$是$[T]$的逆。反之也有

$$[X] = [T]^{-1}[Y] \tag{5-99}$$

也就是说在编码端利用变换得到$[Y]$，在译码端可用反变换来恢复$[X]$。

如果在传输或存储中只保留M个分量，$M<N$，则可由$[Y]$的近似值$[\dot{Y}]$来恢复$[X]$。

$$[\dot{Y}]' = [Y_0, Y_1, Y_2, \cdots, Y_{M-1}]$$

$$[\dot{X}] = [T]^T[\dot{Y}] \tag{5-100}$$

当然$[\dot{X}]$是$[X]$的近似值。但是只要选取得当，仍可保证失真在允许的范围内。

显然，关键问题在于选取什么样的正交变换$[T]$，才能既得到最大的压缩率，又不造成严重的失真。因此，有必要研究一下由正交变换得到$[Y]$的统计特性。$[Y]$的统计特性中最为重要的是协方差矩阵。下面讨论一下正交变换后得到的$[Y]$的协方差矩阵采用何种形式。

设图像信号为一个N维向量：

$$[X] = [X_0, X_1, X_2, \cdots, X_{N-1}] \tag{5-101}$$

当然，$[X]$的统计特性可以测得。

$[X]$的协方差矩阵：

$$[C_X] = E\{([X] - [\overline{X}])([X] - [\overline{X}])^T\} \tag{5-102}$$

式中$[C_X]$是$[X]$的协方差矩阵，$[\overline{X}]$是$[X]$的均值，E是求数学期望值。

又设变换系数向量为

$$[Y]^T = [Y_0, Y_1, Y_2, \cdots, Y_{N-1}] \tag{5-103}$$

$[C_Y]$为$[Y]$的协方差矩阵，所以

$$[C_Y] = E\{([Y] - [\overline{Y}])([Y] - [\overline{Y}])^T\} \tag{5-104}$$

式中$[\overline{Y}]$是$[Y]$的均值。

由正交变换的定义，有

$$[Y] = [T][X]$$

$$[X] = [T]^T[Y]$$

因此

$$\begin{aligned}[C_Y] &= E\{([Y] - [\overline{Y}])([Y] - [\overline{Y}])^T\} \\ &= E\{([T][X] - [T][\overline{X}])([T][X] - [T][\overline{X}])^T\} \\ &= E\{[T]([X] - [\overline{X}])([X] - [\overline{X}])^T[T]^T\}\end{aligned}$$

$$= [T]E\{([X]-[\overline{X}])([X]-[\overline{X}])^T\}[T]^T$$
$$= [T][C_X][T]^T$$

即

$$[C_Y]=[T][C_X][T]^T \tag{5-105}$$

式(5-105)说明，变换系数的协方差矩阵可以通过空间域图像的协方差矩阵的二维变换得到。由此可以得出结论：变换系数的协方差矩阵决定于变换矩阵$[T]$和空间域图像的协方差矩阵$[C_X]$。而$[C_X]$是图像本身所固有的，因此，关键在于寻求合适的$[T]$。

如果$[C_Y]$是一个对角形矩阵，那就说明系数间的相关性完全解除了。也就是说解除了包含在相关性中的冗余度，为无失真压缩编码打下了基础。另外，还希望对角形矩阵中元素的能量尽量集中，以便使舍去若干系数后造成的误差不致于太大，这样，就为熵压缩编码提供了条件。综上所述，变换编码要解决的关键问题是合理地寻求变换矩阵$[T]$。

3. 最佳变换问题

在研究各种变换矩阵$[T]$的过程中，自然要比较它们的优劣，因此，就有一个比较准则问题。下面讨论最佳变换问题。

(1) 最佳变换应满足的条件

最佳变换应满足下面两个条件：

①能使变换系数之间的相关性全部解除；

②能使变换系数之方差高度集中。

显然，第一个条件希望变换系数的协方差矩阵为对角形矩阵；第二个条件希望对角形矩阵中对角线上的元素能量主要集中在前 M 项上，这样就可以保证在去掉 $N-M$ 项后的截尾误差尽量小。

(2) 最佳的准则

常用的准则仍然是均方误差准则。均方误差由下式表示：

$$\overline{e^2}=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}e^2(x,y)$$
$$=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[g(x,y)-f(x,y)]^2 \tag{5-106}$$

式中 $f(x,y)$代表原始图像，$g(x,y)$为经编译码后的恢复图像。均方误差准则就是要使$\overline{e^2}$最小。$\overline{e^2}$最小的变换就是最佳变换。

(3) 均方误差准则下的最佳统计变换

均方误差准则下的最佳统计变换也叫 K-L 变换(Karhunen Loeve Transform)。

设 $\boldsymbol{T}$ 是一正交变换矩阵

$$\boldsymbol{T}^T=[\phi_1\phi_2\cdots\phi_N] \tag{5-107}$$

这是一个 $N\times N$ 矩阵，其中 ϕ_i 是一个 N 维向量。这个矩阵是正交的，因此

$$\phi_i^T\phi_j=\begin{cases}1 & i=j\\0 & i\neq j\end{cases} \tag{5-108}$$

显然
$$\boldsymbol{T}^T\boldsymbol{T}=\boldsymbol{I}$$

另外，设有一数据向量

$$\boldsymbol{X}^T=[X_1X_2\cdots X_N]$$

经正交变换后，

$$X = TY \tag{5-109}$$

这里
$$Y^T = [Y_1 Y_2 \cdots Y_N]$$

而

$$X = T^T Y = [\phi_1 \phi_2 \cdots \phi_N] Y \tag{5-110}$$

于是

$$X = Y_1 \phi_1 + Y_2 \phi_2 + \cdots + Y_N \phi_N = \sum_{i=1}^{N} Y_i \phi_i \tag{5-111}$$

为了压缩数据，在恢复 X 时不取完整的 N 个 Y 分量，而是仅取 M 个分量，其中 $M<N$。这样其中 M 个分量构成一个子集，即

$$[Y_1 Y_2 \cdots Y_M]$$

用这 M 个分量去估计 X，其余的用常量 b_i 来代替。于是可得到：

$$\dot{X} = \sum_{i=1}^{M} Y_i \phi_i + \sum_{i=M+1}^{M} b_i \phi_i \tag{5-112}$$

这里 $\dot{X}$ 是 X 的估计。X 的值与 $\dot{X}$ 的误差为

$$\begin{aligned} \Delta X &= X - \dot{X} \\ &= X - \sum_{i=1}^{M} Y_i \phi_i - \sum_{i=M+1}^{N} b_i \phi_i \\ &= \sum_{i=M+1}^{N} (Y_i - b_i) \phi_i \end{aligned} \tag{5-113}$$

设 ε 为均方误差，则

$$\varepsilon = E\{ \| \Delta X \|^2 \} = E\{ (\Delta X)^T (\Delta X) \} \tag{5-114}$$

将 ΔX 代入 ε，

$$\varepsilon = E\Big\{ \sum_{i=M+1}^{N} \sum_{j=M+1}^{N} (Y_i - b_i)(Y_j - b_j) \phi_i^T \phi_j \Big\}$$

由上述的正交条件可简化为

$$\varepsilon = \sum_{i=M+1}^{N} E\{ (Y_i - b_i)^2 \} \tag{5-115}$$

根据最小均方误差准则，要使 ε 最小就要正确选取 b_i 及 ϕ_i。为了求得最佳的 b_i 和 ϕ_i，可分下面两步来求。

第一步把 ε 对 b_i 求导并令其等于零，即

$$\begin{aligned} \frac{\partial \varepsilon}{\partial b_i} &= \frac{\partial}{\partial b_i} E\{ (Y_i - b_i)^2 \} \\ &= -2[E\{Y_i\} - b_i] = 0 \\ b_i &= E\{Y_i\} \end{aligned} \tag{5-116}$$

又因为
$$Y_i = \phi_i^T X$$

所以，

$$b_i = \phi_i^T E\{X\} = \phi_i^T \overline{X} \tag{5-117}$$

式中 $\overline{X} = E\{X\}$

将 $Y_i = \phi_i^T X$，$b_i = \phi_i^T X$ 代入 ε，则

$$\varepsilon = \sum_{i=M+1}^{N} E\{ (Y_i - b_i)(Y_i - b_i)^T \}$$

$$= \sum_{i=M+1}^{N} \phi_i^T E\{(X-\overline{X})(X-\overline{X})^T\}\phi_i \tag{5-118}$$

因为

$$E\{(X-\overline{X})(X-\overline{X})^T\}=[C_X]$$

所以

$$\varepsilon = \sum_{i=M+1}^{N} \phi_i^T[C_X]\phi_i \tag{5-119}$$

第二步求最佳化的 ϕ_i。为了求得最佳的 ϕ_i，不仅要找出 ϕ_i 使 ε 最小，而且还要满足 $\phi_i^T\phi_i=1$ 的条件。因此，可用求条件极值的拉格朗日乘数法法则。根据拉格朗日乘数法，在求 ϕ_i 的条件极值时做一个新的函数 $\hat{\varepsilon}$：

$$\hat{\varepsilon}= \varepsilon - \sum_{i=M+1}^{N} \beta_i[\phi_i^T\phi_i - 1]$$

$$= \sum_{i=M+1}^{N} \{\phi_i^T C_X\phi_i - \beta_i[\phi_i^T\phi_i - 1]\} \tag{5-120}$$

对 ϕ_i 求导，并注意到

$$\frac{\partial}{\partial\phi_i}[\phi_i^T C_X\phi_i] = 2C_X\phi_i \tag{5-121}$$

$$\frac{\partial}{\partial\phi_i}[\phi_i^T\phi_i] = 2\phi_i \tag{5-122}$$

所以，

$$\frac{\partial\hat{\varepsilon}}{\partial\phi_i} = 2C_X\phi_i - 2\beta_i\phi_i = 0 \tag{5-123}$$

即

$$C_X\phi_i = \beta_i\phi_i \tag{5-124}$$

由线性代数理论可知，

$$\beta_i\phi_i - C_X\phi_i = 0$$

$$(\beta_i E - C_Y)\phi_i = 0 \tag{5-125}$$

显然，β_i 就是 C_X 的特征根，ϕ_i 就是 C_X 的特征向量。如 C_X 是对称矩阵，就可找到一个变换 $[T]$，使 C_Y 成为对角形矩阵。

在关于图像的统计特性的讨论中曾提到，如果图像信源是一阶马尔可夫模型的话，那么 C_X 将是一个 Toeplitz 矩阵，即

$$C_X = \sigma_r^2 \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{N-1} \\ \rho & 1 & \rho & \cdots & \rho^{N-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \cdots & 1 \end{bmatrix} \tag{5-126}$$

这是一个对称矩阵。因此，通过正交变换可以使 C_Y 成为对角形矩阵。也就是说可以找到一个变换矩阵 $[T]$ 而得到最佳变换结果。这就是 K-L 变换的核心。

(4) 最佳变换的实现方法

由上面的分析可见，K-L 变换中的变换矩阵 $[T]$ 不是一个固定的矩阵，它必须由信源来确定。当给定一信源时，可用如下几个步骤求得 $[T]$：

①给定一幅图像后，首先要统计其协方差矩阵 C_X；

②由 C_X 求 λ 矩阵，即 $[\lambda E-C_X]$。并且由 $|\lambda E-C_X|=0$ 求得其特征根，进而求得每一个特征根所对应的特征向量；

③由特征向量求出变换矩阵[T]；

④用求得的[T]对图像数据进行正交变换。

经过上面四步运算就可以保证在变换后使[$\boldsymbol{C}_Y$]=[T][$\boldsymbol{C}_X$][T]^{T}是一个对角形矩阵。这个[T]就是 K-L 变换中的变换矩阵。

通过上面的讨论不难看出，图像不同，$\boldsymbol{C}_X$ 就不同，因此[T]也就不同。为了得到最佳变换，每送一幅图像就要重复上述四个步骤，找出[T]后再进行正交变换操作，所以运算相当繁琐，而且没有快速算法。此外，K-L 变换在数学推导上总能实现，但用硬件实现就不那么容易了。因此，K-L 变换的实用性受到很大限制，一般多用来作变换性能的评价标准。当然，寻求 K-L 变换的简洁算法也是许多人研究的课题。下面举一个简单的例子说明 K-L 变换的具体实现方法。

例：已知某信源的协方差矩阵为 $\boldsymbol{C}_X$，求最佳变换矩阵[T]。

$$\boldsymbol{C}_X = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

写出 $\boldsymbol{\lambda}$ 矩阵：

$$[\boldsymbol{\lambda} E - \boldsymbol{C}_X] = \begin{bmatrix} \lambda - 1 & -1 & 0 \\ -1 & \lambda - 1 & 0 \\ 0 & 0 & \lambda - 1 \end{bmatrix}$$

$$f(\lambda) = \begin{vmatrix} \lambda - 1 & -1 & 0 \\ -1 & \lambda - 1 & 0 \\ 0 & 0 & \lambda - 1 \end{vmatrix}$$

$$= (\lambda - 1)^3 - (\lambda - 1) = 0$$

求得 $\lambda_1=2$，$\lambda_2=1$，$\lambda_3=0$。

求 $\lambda_1=2$，$\lambda_2=1$，$\lambda_3=0$ 的特征向量：

$$(\lambda_1 E - \boldsymbol{C}_X)X = 0$$

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = 0$$

$$\begin{cases} X_1 - X_2 = 0 \\ -X_1 + X_2 = 0 \\ X_3 = 0 \end{cases}$$

所以，其基础解系为（1 1 0），归一化后为

$$\left(\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad 0 \right)$$

同理，$\lambda_2=1$ 时特征向量为 $(0 \quad 0 \quad 1)$

$\lambda_3=0$ 时特征向量为 $\left(\frac{1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \quad 0 \right)$

由上面的结果可求得[T]：

$$[T] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \end{bmatrix}$$

$$[T]^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \\ 0 & 1 & 0 \end{bmatrix}$$

$[T]$便是 K-L 变换的变换矩阵。

4. 准最佳变换

最佳变换的性能固然好，但实现起来却不容易。因此，在实践中更加受到重视的是一些所谓的准最佳变换。什么是准最佳变换呢？最佳变换的核心在于经变换后能使$[C_Y]$成为对角形矩阵形式。如果能找到某些固定的变换矩阵$[T]$，使变换后的$[C_Y]$接近于对角形矩阵，那也是比较理想的了。

在线性代数理论中知道，任何矩阵都可以相似于一个约旦形矩阵，这个约旦形矩阵就是准对角形矩阵，其形状如式(5-127)所示：

$$\begin{bmatrix} \lambda_0 & & & & & \\ 0 & \lambda_1 & & & & \\ & 1 & \lambda_2 & & 0 & \\ & & 0 & \ddots & & \\ & & & \ddots & \lambda_{N-2} & \\ 0 & & & & 1 & \lambda_{N-1} \end{bmatrix} \tag{5-127}$$

其主对角线上是特征值，在下对角线上仅有若干个 1，这也就比较理想了。从相似变换理论可知，总可以找到一个非奇异矩阵 $\boldsymbol{T}$，使得

$$\boldsymbol{T}^T A \boldsymbol{T} = B \tag{5-128}$$

而且这个 $\boldsymbol{T}$ 并不是唯一的。

在第 3 章讨论过的五种正交变换都具有 $\boldsymbol{T}$ 的性质。这五种正交变换就是常用的准最佳变换。尽管它们的性能比 K-L 变换稍差，但是，由于它们的变换矩阵 $\boldsymbol{T}$ 是固定的，这给工程实现带来了极大的方便。因此，首先付诸于实用的是这些准最佳变换法。

5. 各种准最佳变换的性能比较

从运算量大小以及压缩效果这两个方面来比较各种正交变换，其性能比较如表 5-16 所示。

表中列举了一维 N 点各种正交变换所需的运算次数。从下至上的顺序代表了从运算量大小或硬件设备量的角度来看的优劣次序。而其压缩效果与表中的箭头方向正好相反。从表中可见，K-L 变换的运算量最大，极难做到用硬件来实现，而沃尔什-哈达玛变换运算量最小，用一般高速 TTL、ECL 数字集成电路就可以做到实时变换。但是其压缩效果则较差。假如图像信号为马尔可夫模型，那么各种正交变换在变换域能量集中优劣情况如下面的顺序：

$$\text{K-L}\rightarrow\text{DCT}\rightarrow\text{ST}\rightarrow\text{DFT}\rightarrow\begin{matrix}\text{DWHT}\\ \text{HT}\end{matrix}$$

优————————————→劣

表 5-16　各种正交变换性能比较

	正交变换类型	运 算 量	对视频图像实时处理的难易程度
劣	K L	求$[C_X]$及其特征值，特征矢量，矩阵运算要N^2次实数加法和N^2次实数乘法	极难做到
	DFT	$N\log_2N$次复数乘法以及$N\log_2N$次复数加法	较复杂
	FDCT	$\frac{3N}{2}\log_2(N-1)+2$次实数加法以及$N\log_2N-\frac{3N}{2}+4$次实数乘法	采用高速CMOS/SOS大规模集成电路，能做到实时处理
	ST	当$N=4$时，共需8次加或减以及6次乘法	可能实时
	DWHT	$N\log_2N$次实数加或减	可利用一般高速TTL，ECL数字集成电路做到实时
优	HT	$2(N-1)$次实数加或减	与DWHT相同

6. 编码

变换为压缩数据创造了条件，压缩数据还要靠编码来实现。通常所用的编码方法有二种，一是区域编码法，一是门限编码法。

(1) 区域编码法

这种方法的关键在于选出能量集中的区域。例如，正交变换后变换域中的能量多半集中在低频率(或列率)空间上，在编码过程中就可以选取这一区域的系数进行编码传送，而其他区域的系数可以舍弃不用。在译码端可以对舍弃的系数进行补零处理。这样由于保持了大部分图像能量，在恢复图像中带来的质量劣化并不显著。

在区域编码中，区域抽样和区域编码的均方误差均与方块尺度有关。图5-52示出了图像变换区域抽样的均方误差与方块尺度的关系。图5-53则示出了图像区域编码均方误差与方块尺度的关系。区域编码的显著缺点是一旦选定了某个区域就固定不变了，有时图像中的能量也会在其他区域集中较大的数值，舍掉它们会造成图像质量较大的损失。

(2) 门限编码

这种采样方法不同于区域编码法，它不是选择固定的区域，而是事先设定一个门限值T。如果系数超过T值，就保留下来并且进行编码传送。如果系数值小于T值就舍弃不用。这种方法有一定的自适应能力。它可以得到较区域编码为好的图像质量。但是，这种方法也有一定的缺点，那就是超过门限值的系数的位置是随机的。因此，在编码中除对系数值编码外，还要有位置码。这两种码同时传送才能在接收端正确恢复图像。所以，其压缩比有时会有所下降。一种简单实用的位置编码技术是对有效样本之间的无效样本数目编码。这种行程长度编码可以按下述方案设计：

1)对每一行的第一个样值不管其幅值如何都编码，并且把全0或全1的位置码字附加到

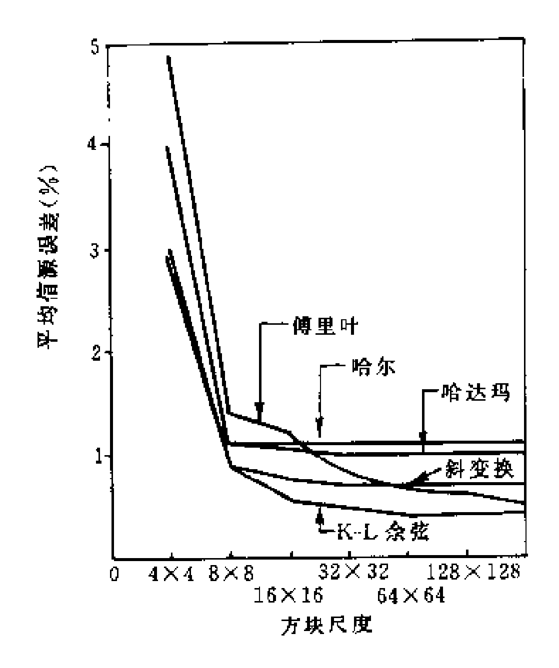


图 5-52 图像变换区域抽样的均方误差与方块尺度的关系

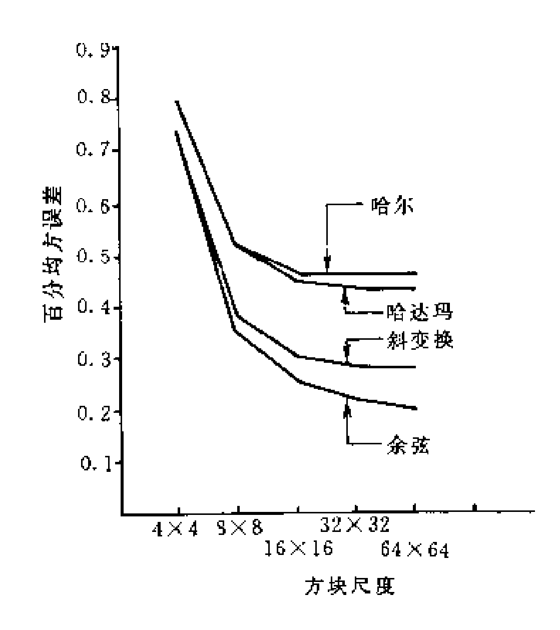


图 5-53 图像区域编码均方误差与方块尺度的关系

这一行程幅度内，作为行同步码。

2)第二个行程长度码字的幅度是下一有效样值的编码幅度，而位置代码是由前一有效样值起进行二进制计数的无效样值数。

3)如果在扫描样值的最大行程长度以后没有出现新的有效样值，那么位置和幅度代码比特都取为 1，以表示这是最大行程长度。

在这个方案中包含有行同步码组，它的优点是不再需要对行数编码，同时可以避免信道误差扩散超出一行。

对于系数值的编码可用变长码，也可用等长码。

利用门限编码法可能得到的压缩比可作如下估计。设 WHT 的变换系数值服从正态分布，则有

$$p[F(k)] = \frac{1}{\sqrt{2\pi\sigma^2(k)}}\exp\left(-\frac{F^2(k)}{2\sigma^2(k)}\right) \tag{5-129}$$

$p[F(k)]$为系数值分布密度，$\sigma^2(k)$为其方差，$F(k)$是变换系数之幅值。设门限值为 $F_T(k)$那么超过门限 $F_T(k)$的概率为

$$\begin{aligned}
p(k) &= \int_{F_T(k)}^{\infty} p[F(k)]\mathrm{d}F(k) \\
&= \int_0^{\infty} p[F(k)]\mathrm{d}F(k) - \int_0^{F_T(k)} p[F(k)]\mathrm{d}F(k) \\
&= \int_0^{\infty} \frac{1}{\sqrt{2\pi\sigma^2(k)}}\exp\left(-\frac{F^2(k)}{2\sigma^2(k)}\right)\mathrm{d}F(k) \\
&\quad - \int_0^{F_T(k)} \frac{1}{\sqrt{2\pi\sigma^2(k)}}\exp\left(-\frac{F^2(k)}{2\sigma^2(k)}\right)\mathrm{d}F(k)
\end{aligned}$$

$$= 0.5 - \frac{1}{\sqrt{2\pi}}\int_0^{F_T(k)} e^{-\frac{F^2(k)}{2\sigma^2(k)}} \cdot \frac{1}{\sigma(k)} dF(k)$$

$$= 0.5 - \frac{1}{\sqrt{2\pi}}\int_0^{F_T(k)} e^{-\frac{\left[\frac{F(k)}{\sigma(k)}\right]^2}{2}} d\left[\frac{F(k)}{\sigma(k)}\right]$$

$$= 0.5 - \mathrm{erf}\left(\frac{F_T(K)}{\sigma(k)}\right)$$

其中

$$\mathrm{erf}(y) = \frac{1}{\sqrt{2\pi}}\int_0^x e^{-\frac{y^2}{2}} dy \tag{5-130}$$

为误差函数。

$$N_T = \sum_{k=0}^{N-1}\left[0.5 - \mathrm{erf}\left(\frac{F_T(k)}{\sigma(k)}\right)\right] \tag{5-131}$$

如果令$\frac{F_1(k)}{\sigma(k)}=r$,那么可能的压缩比为

$$\frac{N}{N_T} = [0.5 - \mathrm{erf}(r)]^{-1} \tag{5-132}$$

其中假定$F_r(k)=r\sigma(k)$呈线性关系。N是系数总数，N_T是超过门限的系数数目。式(5-132)就是门限编码法可能得到的压缩比估计。

7. 一种实时电视信号 WHT 编码方案

正交变换编码法与其他编码方法相比有明显的优点。例如，在获得相同质量的图像时变换编码能得到更大的压缩；变换编码所产生的失真比较容易为人们所接受；变换编码对图像的统计特性的变化不太敏感，同时误码没有扩散性。它的主要缺点是设备比较复杂，编码延时比较大。但是随着高速数字器件的发展，这一缺点并不是一个难以愈越的障碍。因此，人们对变换编码进行了广泛的研究，寻找各种变换方法加以模拟，从中选取最佳方案。下面介绍一个电视信号实时沃尔什-哈达玛变换压缩编码装置。

(1) 沃尔什-哈达玛变换的主要优点

沃尔什-哈达玛变换的原理及性质已在第 3 章中做了较详细讨论。沃尔什-哈达玛变换除了具有一般正交变换所具有的优点，即能量保持、熵保持、去相关、重新分布和集中能量外，还有一些其他优点：

1) 变换算子是由+1 和−1 组成的哈达玛矩阵，这个矩阵很容易由递推关系得到 2^N 阶矩阵。

2) 它是一实对称矩阵，因此，在运算中只有实数运算而没有复数运算。同时，正、反变换均用同一矩阵形式，这就可以用相同的电路来实现正、反变换，给电路设计带来很大的方便。

3) WHT 对实时处理十分有利。实时处理的关键是速度，它必须在预定的时间内完成处理任务，否则就会造成极大的混乱。由于 WHT 只有加减运算，没有乘除运算，这样其运算速度非常快。用快速算法只要 $N\log_2 N$ 次加减法就可完成一次变换。数字电视信号是一个连续不断的高速实时序列，用一维变换装置来处理高速序列必须在 N 个取样间隔内算出变换系数。这样才能有条不紊地处理源源涌入的数据流。因此，运算速度决定着能否进行实时处理。在这一点上，WHT 较之其他变换有很大优点。

4) WHT 的硬件设备量较其他变换为小，这也是它的一个突出优点。

(2) 实时 WHT 的硬件设计

通常 WHT 的硬件可根据快速算法流程图采用存储器来设计。这种方案需要能高速写入和读出的存储器。对于电视信号进行实时变换的要求就更高。因此，这里没有采用这种方案，而是采用流水作业的方案用小规模电路来实现实时变换的。

这个装置采用 4 阶 WHT，其运算过程如下（用十进制运算说明）。

设输入信号序列为$[f(X)]$，

$$[f(X)] = [X_1X_2X_3X_4]$$

$$[F]=\frac{1}{4}[X_1X_2X_3X_4]\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

$$=[(X_1+X_2+X_3+X_4)(X_1-X_2+X_3-X_4)$$
$$(X_1+X_2-X_3-X_4)(X_1-X_2-X_3+X_4)]$$
$$=Y_1Y_2Y_3Y_4$$

所以，

$$Y_1=\frac{1}{4}(X_1+X_2+X_3+X_4)$$

$$Y_2=\frac{1}{4}(X_1-X_2+X_3-X_4)$$

$$Y_3=\frac{1}{4}(X_1+X_2-X_3-X_4)$$

$$Y_4=\frac{1}{4}(X_1-X_2-X_3+X_4)$$

反变换如下：

$$[f(Y)] = [Y_1Y_2Y_3Y_4]$$

$$[f(X)]=[Y_1Y_2Y_3Y_4]\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

$$=[(Y_1+Y_2+Y_3-Y_4)(Y_1-Y_2+Y_3-Y_4)$$
$$(Y_1+Y_2-Y_3-Y_4)(Y_1-Y_2-Y_3+Y_4)]$$

即

$$X_1=Y_1+Y_2+Y_3+Y_4$$
$$X_2=Y_1-Y_2+Y_3-Y_4$$
$$X_3=Y_1+Y_2-Y_3-Y_4$$
$$X_4=Y_1-Y_2-Y_3+Y_4$$

根据上面的运算过程可直接设计出 WHT 的流水作业硬件。其原理方框图如图 5-54 所示。图 5-54 的原理框图可由如下等效电路来说明，第一级运算等效电路如图 5-55 所示。图中 T 是移位寄存器，由它得到步进延时。如果输入信号依次是 $X_1,X_2,X_3,\cdots$，那么其运算流程时序图如图 5-57 所示。由流程时序图可见，第一级运算得到一个$[u]$序列。$[u]$序列送至第二级运算就可以得到变换系数序列，第二级运算等效电路见图 5-56。它的运算时序图示于图 5-58。

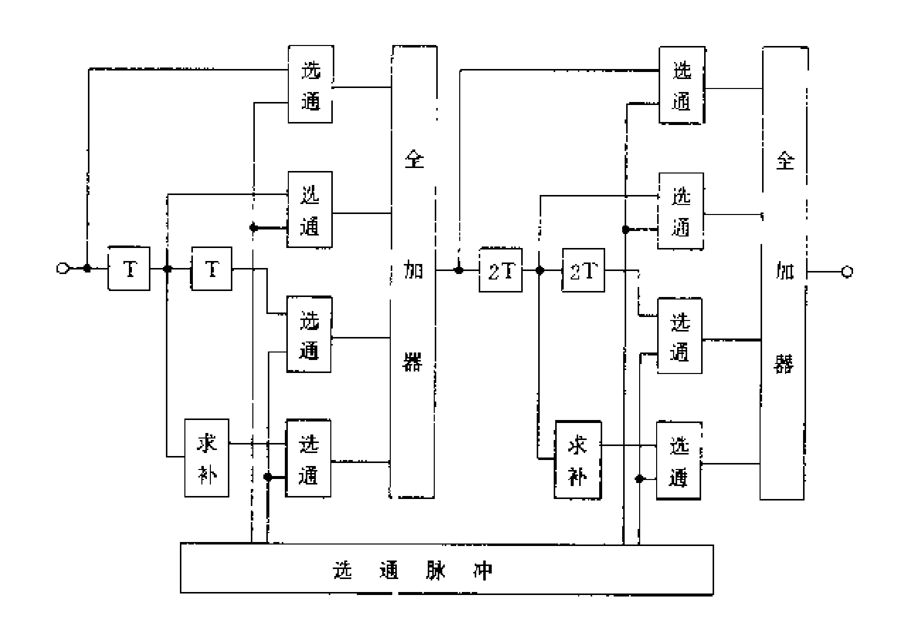


图 5-54　WHT 原理框图

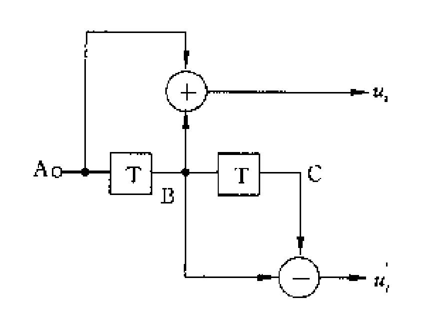


图 5-55　第一级运算等效电路

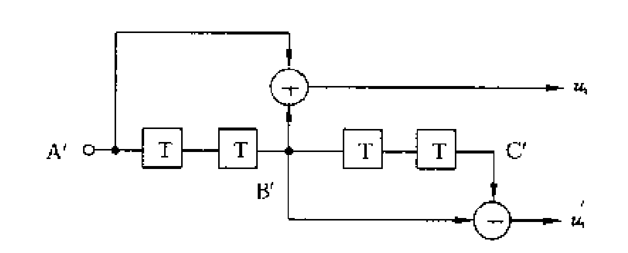


图 5-56　第二级运算等效电路

5.7　图像编码的国际标准

随着计算机网络及非话通信业务的迅速发展，图像通信已越来越受到全世界科技工作者的关注。以往的非标准的工作状态极大地制约了图像处理技术的发展与应用。因此，CCITT、ISO、IEC 等国际组织积极致力于图像处理的标准化工作。特别是图像编码，由于它涉及到多媒体、HDTV、数字电视、可视电话、会议电视等图像传输方面的广泛应用，所以，相关的国际组织成功地制定了一系列的国际标准，极大地推动了图像编码技术的发展与应用。目前，众所周知的一些编码标准有：JPEG、MPEG、JBIG 及 H. 26x 等标准。

1) JPEG 标准　在 1986 年，ISO 和 CCITT 成立了"联合图片专家组"(Joint Photographic Expert Group)，他们的主要任务是研究静止图像压缩算法的国际标准。1987 年用 Y：U：V＝4：2：2，每像素 16bit，宽高比为 4：3 的电视图像进行了测试，遴选出三个方案，进行评选，其中，8×8 的 DCT 方案得分最高，经细致的工作，于 1991 年 3 月提出 ISO CD 10918 号建议草案，这就是 JPEG 标准。

2) MPEG 标准　ISO 和 CCITT 于 1988 年成立了"活动图像专家组"(Moving Picture Expert Group)。该组的任务是制定用于数字存储媒介中活动图像及伴音的编码标准。于 1991 年 11 月提出了 1.5Mb/s 的编码方案。1992 年通过了 ISO 11172 号建议，这就是 MPEG 标准，

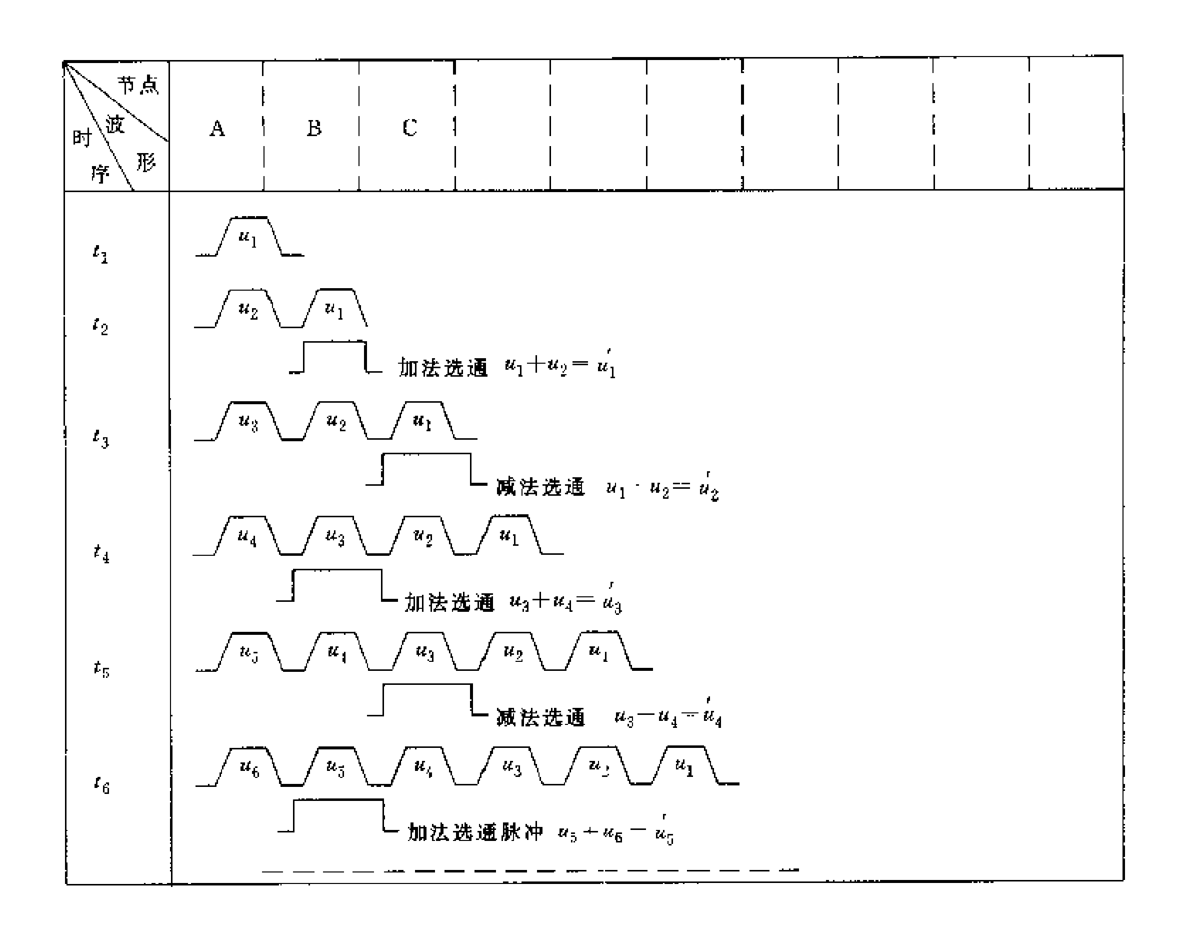


图 5-57 第一级运算时序图

后来的发展又出现了 MPEG2、MPEG4、MPEG7 等针对不同应用的多种标准，因此，把最初的 MPEG 标准定为 MPEG1 标准。

3) JBIG 标准 1988 年，由 ISO-IEC/JTC1/SC2/WG9 和 CCITT 第 13 研究组成立了“联合二值图像专家组”(Joint Bi-level Image Expert Group)，它的主要任务是研究制定用于二值图像的编码标准。于 1991 年 10 月提出 ISO/IEC CD 11544(CCITT T.82)号建议草案，这就是二值传真图像压缩标准。

4) H.261 标准 为适应可视电话和会议电视的需要，CCITT 第 15 研究组承担了视频的编解码标准的研究工作。并于 1988 年提出了视频 CODEC 的 H.261 建议。1990 年通过了 $P\times$ 64kb/s 的编码方案，其中 $P=1\sim30$。1990 年 12 月在日内瓦最后通过了该建议。从这些编码标准来看，初步解决了静止图像、可视电话/会议电视、多媒体视频乃至 HDTV 的压缩编码的需要。从所采用的技术来看，都采用了最基本的编码技术，通过组合应用，达到了预期的编码效果。因此，这些编码方法都属于混合编码的范畴。下边我们就这些编码标准，对典型的标准采用的具体技术和编码原理进行一些简单的介绍，以便供从事该项工作的技术人员参考。

5.7.1 H.261 编码标准

1. 编码方案的提出

可视电话和会议电视由于其实用性受到人们的广泛注意。目前，它在非话业务中占有重要地位

早在 1984 年 CCITT 的第 15 研究组专门研究可视电话的编码问题。提出了一个 H.120

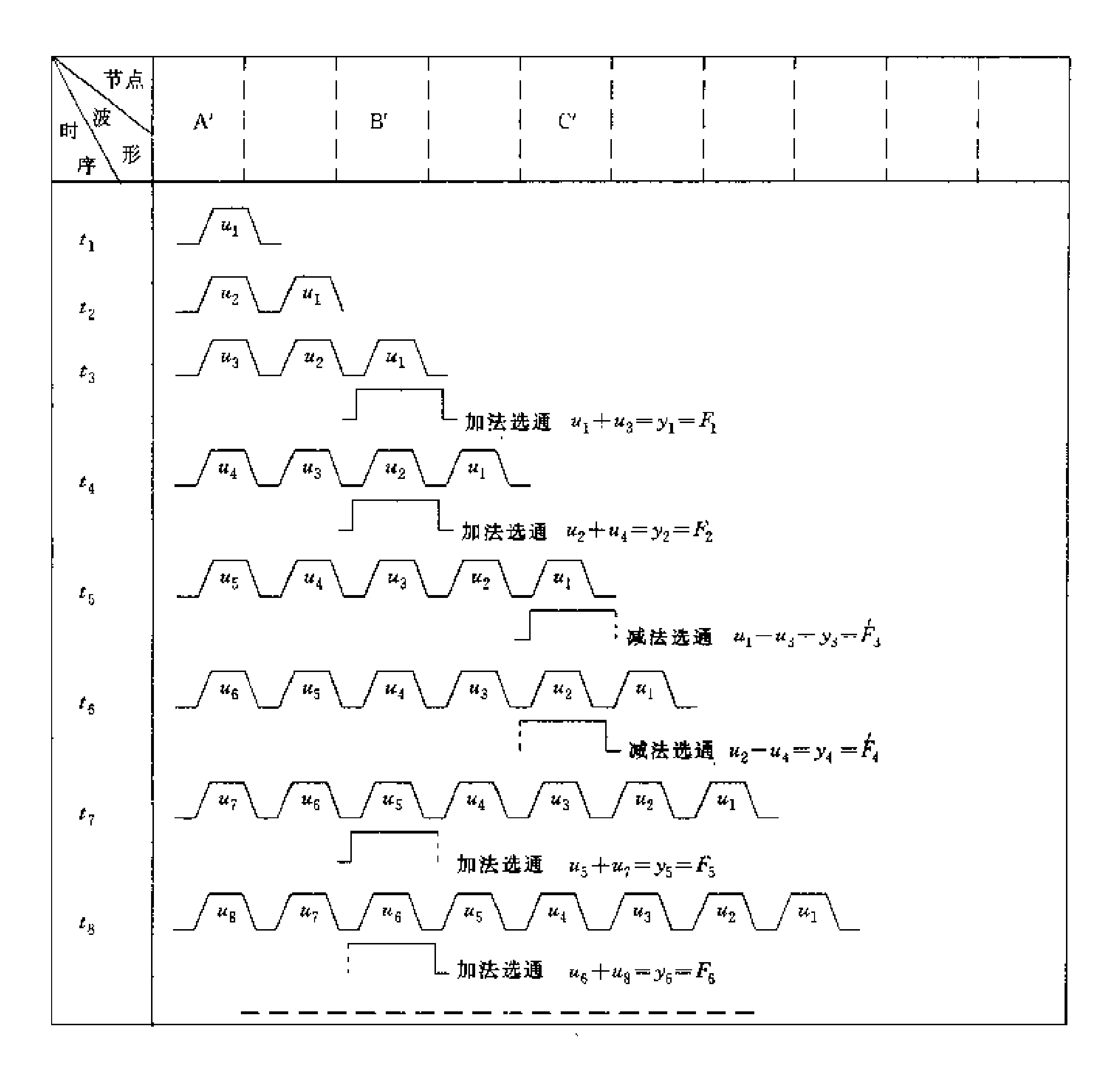


图 5-58 第二级运算时序图

建议。针对 625 行/帧，50 场/秒，在 PCM 一次群上传图像信号，1988 年提出了一个传输速率为 5 级的标准，$P\times384$kb/s，$P=1,2,3,4,5$。384kb/s 在 ISDN 中称为 H_0 通道，后来由于 384kb/s 起点偏高，跨度也大，灵活性受到影响，因此，在 1990 年提出了 $P\times64$kb/s，$P=1,2,3,\cdots,30$ 的标准，这就是在 1990 年 12 月批准的 H.261 建议。用于可视电话时 $P=2$，速率只有 128kb/s，当用于会议电视时，建议 $P\geqslant6$，速率 384kb/s，最高可达 2048kb/s。

2. H.261 的图像格式

图像的纵横像素数是图像的基本格式。为了使现行的各种电视制式(PAL、NTSC、SECAM)方便地转换为电视会议和可视电话的图像形式，所以，H.261 标准采用通用的中等格式，即 CIF (Common Intermediate Format) 格式或通用的 1/4 中等格式 QCIF (Quarter Common Intermediate Format)格式。

由于彩色电视是采用相加混色原理实现的，所以，彩色电视信号遵循三基色原理，即：所有颜色都可以由红、绿、蓝三基色混合得到。这就是著名的 Grassman 定律。相加混色的规律符合 $Y=0.30R+0.59G+0.11B$(相加混色)的公式。

通常，彩色信号编码不针对 R、G、B 三原色，而是针对 Y 亮度，U、V 色差信号进行处理。这里：

$$U = B - Y \qquad V = R - Y \tag{5-133}$$

或色调 H(Hue),饱和度 S(Saturation),亮度 I (Intensity)。在 H.261 标准中规定:对于 Y 分量、横向 352 像素,纵向 288 像素。对于色度分量 U 和 V 分别为横向 176 像素,纵向 144 像素。图像尺寸纵横比 4∶3,与普通电视一样。由此,可由图像尺寸的纵横推出:

$$\text{像素纵横比} = \frac{3}{288} : \frac{4}{352} = 11 : 12 \tag{5-134}$$

近于方形。

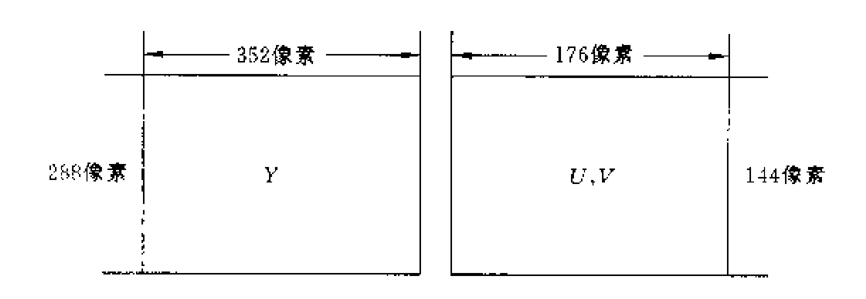


图 5-59　CIF 的 Y 分量格式及 CIF 的 U、V 分量格式

由图(5-59)可见,Y、U、V 像素的面积相同,但像素数并不相同,所以,色度分量的清晰度低于 Y 分量,由于人的视觉特性,这一差别并不影响视觉清晰度。此外,考虑到当前的电视制式分别是 625 行/帧、25 帧/秒和 525 行/帧、30 帧/秒,都是隔行扫描,1 帧等于 2 场,故两种制式的场扫描行数为 625/2 和 525/2。288 是从这两种扫描行数转换来的,即取两种扫描行的平均值:

$$\left(\frac{625}{2} + \frac{525}{2}\right)/2 = 287.5 \approx 288 \tag{5-135}$$

这样,便于实现 CIF 格式与两种制式的相互转换,同时,也符合两制式均衡负担的国际原则。由于编码是以 8×8 像素块作为基本单元,所以纵横像素数都应是 8 的整数倍。即:

$$\frac{352}{8} = 44 \qquad \frac{288}{8} = 36 \qquad \frac{176}{8} = 22 \qquad \frac{144}{8} = 18 \tag{5-136}$$

所以,亮度分量 Y 的 8×8 块数为 1584 块,色度分量 U、V 为 396 块,亮度分量是色度分量的 4 倍,故尔 4 个亮度分块和两个色度分量块共 6 块组成同一个分区,形成 MB,以便于编码。

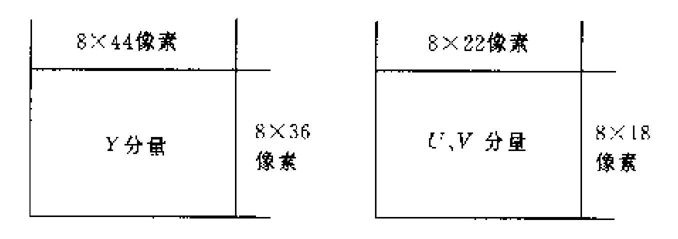


图 5-60　Y 分量与 U、V 分量

在可视电话中由于 $P=1$ 或 2,最高比特率就是 128kb/s,CIF 的像素太多,所以取一半,即:QCIF(Quarter Common Intermediate Format),该格式为:Y 分量横向 176 像素,纵向为 144 像素。色度分量为横向 88 像素,纵向 72 像素。QCIF 是可视电话的最低要求,可视电话均应达到这一要求。而 CIF 规格则是可选的。

在 H.261 标准中,Y、U、V 的建立应符合 CCIR-601(国际无线电咨询委员会)文件的要求,其计算公式为

$$\begin{bmatrix} Y \\ U-128 \\ V-128 \end{bmatrix} = \begin{bmatrix} \frac{77}{256} & \frac{150}{256} & \frac{29}{256} \\ \frac{-44}{256} & \frac{-87}{256} & \frac{131}{256} \\ \frac{131}{256} & \frac{-110}{256} & \frac{-21}{256} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5-137}$$

式中分母取 256 是因为 R、G、B 信号是 8 位量化，有 256 个电平，式中色度信号减 128 是把零电平上移 128，即总电平的一半，主要是色度信号经常有正有负，这里先上移，以后在 DCT 变换前和亮度信号 Y 一起下移，色度信号恢复原来电平，对降低码率有利。按 CCIR-601 的规定，量化后的色度信号峰峰值为 224 电平，最低电平为 16，最高电平为 240，亮度最高电平为 220，最低电平为 16。通常可视电话一般 10 帧/秒，会议电视帧 25 帧/秒。

3. 图像信号的编解码

在 H.261 标准中图像信号编解码方框图如图 5-61 所示。

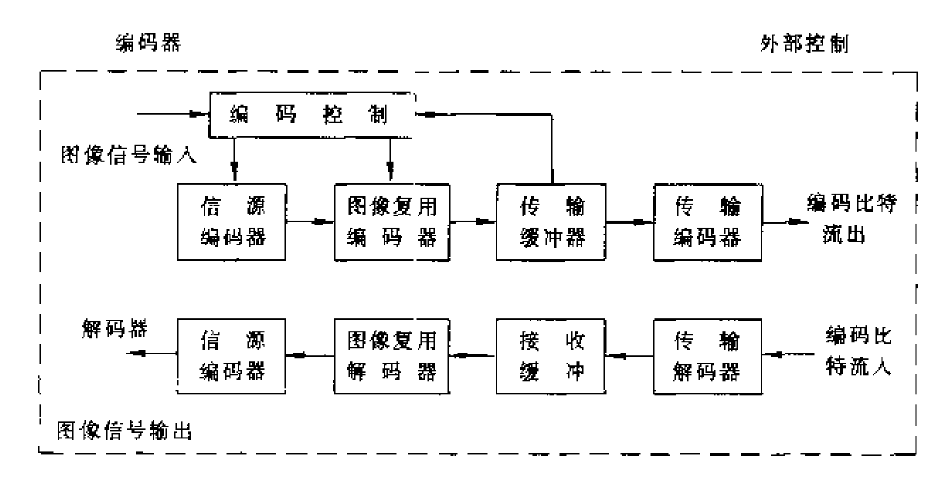


图 5-61 H.261 编解码方框图

图像信号输入、输出指的是 CIF 或 QCIF 格式的数字信号，如果是 NTSC、PAL 或 SECAM 信号应先解成 R、G、B 信号经模数转换，再变换为 Y、U、V 亮度及色度信号，然后再转换为 CIF 或 QCIF 格式和帧频 30Hz 的信号，经帧存储器缓冲后进入上图的输入端。左端输出仍是 CIF 或 QCIF 格式、帧频 30Hz 信号，经相反的变换，还原成视频复合信号。图中的比特流可进入 ISDN 网或其他信道。

信源编码器的作用主要是压缩，采用 DCT 变换后把系数量化，之后输入到图像复用编码器。图像复用编码器的功能是把每帧图像数据编排成 4 个层次的数据结构，同时对交流 DCT 系数进行可变长度编码(VLC)，对直流 DCT 系数进行固定长度编码(FLC)。编码位流送入传输缓冲器。传输缓冲器的存储量是按比特率 $P\times 64$kb/s 加上固定富余量后确定的。由于图像内容变更使传输比特率变更，可以在缓冲器中得到反映。由此，传给上方编码控制 CC 方框，由 CC 控制信源编码中量化器的步长，同时把步长辅助数据送到图像复用编码中的相应层次，以供解码用。这样就可以自动控制比特率的高低。以便适应图像变更的内容，充分发挥即定的比特率 $P\times 64$kb/s 的传输能力，即尽可能保持比特率满负载。

传输编码器主要功能是插入 BCH 正向纠错码，以便传输终端的解码器能检测和纠正错误码字。H.261 中规定要用 BCH 纠错码，在解码中可任选。另外传输编码中还要插入同步码，以便解码器正确解码。

图中的编码控制器，除控制量化步长外，还控制编码模式，即控制编码应是帧间编码还是帧内编码。这一操作在信源编码中进行。图中的外部控制有两个功能，即：

① CIF 和 QCIF 的格式选择；

② 允许每二帧图像之间有零到三帧图像不传。这主要是电视电话的帧间相关性强，不传的图像可由已传的图像计算得到。这属于帧间编码。

4. 信源编码

信源编码的方框图如图 5-62 所示。

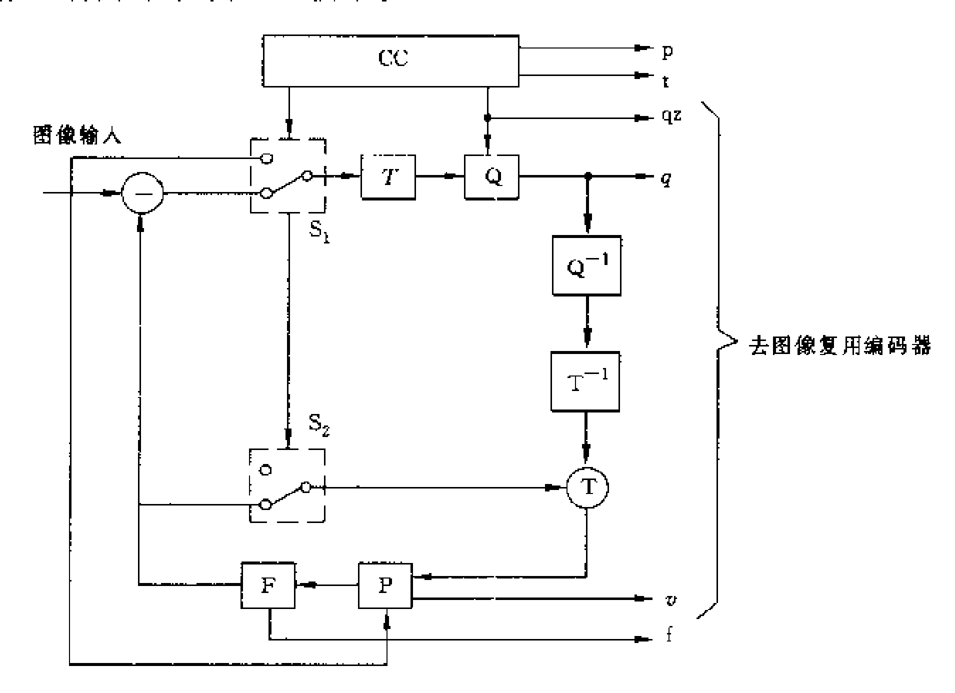


图 5-62 信源编码方框图

图中符号的意义如下：

T—变换　　p—帧内帧间指示

Q—量化器　　t—图像数据传输与否指示

P—帧存储器，运动补偿可变延迟　　qz—量化步长指示

q—量化变换系数

F—环路滤波器　　v—运动矢量

CC—编码控制　　f—环路开关滤波器指示

图 5-62 中，图像输入是以宏块 MB 为单位输入的，MB 中包含亮度 Y 的 4 个 8×8 像素块，色度 U、V 的各一个 8×8 块，共 6 个 8×8 块。

(1) 帧内帧间编码

可视电话帧频是 30 帧/秒，由于帧间的强相关性，所以允许每两帧传送图像之间可以有 3 帧图像不传。每次场景更换后，第 1 帧一定要传，所以对第 1 帧做帧内编码，所传的叫帧内帧 I (Intraframe)表示(见图 5-63)。

第 5 帧为预测帧，用 P 表示，它由第 1 帧和第 5 帧本身经预测编码得到的。P 帧也可以做为下一个 P 帧预测编码的基础。P 帧称为双向内插帧，它由相邻的 I、P 或 P、P 帧计算得到的。因此，I 帧和 P 帧是产生全部 B 帧基础。通常每 12 帧或 15 帧传 1 帧 I 帧，每 3 帧或 4 帧传 1 帧 P 帧。换场景后第 1 帧为 I 帧。

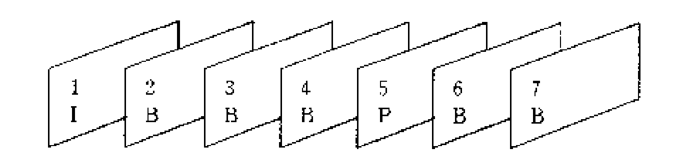


图 5-63 帧间编码

计算公式：第 2 帧(B)

$$B=\frac{3}{4}I+\frac{1}{4}P \tag{5-138}$$

第 3 帧(B)

$$B=\frac{1}{2}(I+P) \tag{5-139}$$

以此类推可计算出其他 B 帧。这种形成 P 帧和 B 帧的方法均称为帧间编码。

图中的开关 S_1,S_2 由 CC 控制，用于帧内帧间编码选择。向下为帧间编码，向上为帧内编码。为了自动决定输入的大块 MB 采用帧内、还是帧间编码，这里确定了一个判决条件。该条件可采用如下方法得到。首先将前帧图像存在帧存储器，后一帧图像来临时，比较前后图像的相关性，如果相关性较弱，则采用帧内编码方法、如果相关性强则采用帧间编码，该判据用于大块 MB。设前帧大块的亮度信号像素值为 $P(x,y)$，后帧图像大块的亮度像素值为 $C(x,y)$，前帧大块亮度信号方差为 V，则

$$V=\frac{\sum_{y=1}^{16}\sum_{x=1}^{16}P(x,y)^2}{256}-\left[\frac{\sum_{y=1}^{16}\sum_{x=1}^{16}P(x,y)}{256}\right]^2 \tag{5-140}$$

在大块内亮度信号有 4 个 8×8 方块（或 16×16 方块），共有 256 个像素，V 实际上是反映前帧图像反差的强弱。前后帧因时间差引起像素差，这里用时间预测变动 VAR 表示，则有

$$\mathrm{VAR}=\frac{\sum_{y=1}^{16}\sum_{x=1}^{16}[C(x,y)-P(x,y)]^2}{256} \tag{5-141}$$

该式就是前后帧对应像素之差的均方值。VAR 说明前后帧像素值变化所导致平均能量的变化。这里，像素取值在 0～256 之间。VAR 值越小，则相关性越大，若考虑到图像反差，则反差大的图像 VAR 也相应的增大。根据 V 和 VAR 的值可以定出如下三条帧内、帧间编码的判据：

1）当 VAR≤64 时为帧间编码模式；

2）当 VAR>64，V≥VAR 时为帧间编码模式：

3）当 VAR>64，VAR>V 时为帧内编码模式。

若采用帧内编码，则对该大块 MB 进行 DCT 变换和量化等后续操作，如果采用帧间编码，则该大块属于 P 帧，要进行运动估计等编码，若采取不传，则该大块属于 B 帧。上述判据是编码程序的一部分。应该说明的是该判据不属于 H. 261 标准，因此，也可以采用其他判据。

(1) 帧内模式

当编码器选择帧内模式时，开关 S_1，S_2 打到上方，输入宏块 MB 数据，MB 经开关输入 DCT 变换方框 T(数据经过下移处理)经 8×8 数据变换，DCT 输出变换系数，并送入量化器 Q，一个量化器对应一个量化步长，该量化步长由缓冲存储器根据存储余量状态告知 CC，再由 CC 传到量化器 Q 及反量化器 Q^{-1}，经量化后的数据，一路从信源编码器输出，进入图像复用编码器，另一路进入反馈环路中的反量化器 Q^{-1}，经反量化器输入到反变换器 T^{-1}，经反变换

后恢复图像数据。再经加法器进入方框 P 中的帧存储器，直到全帧存完为止。

（2）帧间模式：

当采用帧间模式时，开关 S_1，S_2 打向下方，这时，P 帧存储器中已存有前帧图像数据。当后帧的宏块到来时将做如下工作：①由 P 中的模式判据公式决定所采取的模式，如果判据公式决定应采取帧间模式，则进入第二步，否则按帧内模式进行。②根据运动估计公式，在后帧 MB 所对应的前帧 MB 的 ±15 个像素范围内搜索最匹配的亮度块，即 4 个 8×8 亮度数据块，找到最佳前帧匹配数据块后，即可得到运动矢量的两个分量 H 和 V，H、V 可送到图像复用编码器去供编码输出。③由运动矢量确定的前帧 4 个 8×8 亮度数据块和 2 个 8×8 色度数据块从 P 的帧存储器中逐块输出数据 $P(x+H,y+V)$，每块数据通过环路滤波器乘上滤波系数后计做 $P_{\mathrm{F}}(x+H,y+V)$，之后，再分两路，一路向上到减法器，与后帧数据块 $Y(x,y)$ 相减得到差值，宏块数据差值记为

$$\Delta MB_{xy}=Y(x,y)-P_{\mathrm{F}}(x+H,y+V),$$

这就是宏块预测误差。该宏块数据经过上开关 S_1 进入 DCT 变换和量化器，输出后分两路，一路到图像复用编码器，另一路进入反馈回路，经过反量化，反 DCT 变换，进入加法器。由环路滤波器输出的另一路数据 $P_{\mathrm{F}}(X+H,y+V)$ 经过下开关与预测误差 $Y(x,y)-P_{\mathrm{F}}(x+H,y+V)$ 在加法器中相加，得到后帧数据 $Y(x,y)$ 存在帧存储器中，直到帧存储器存完，再开始下一帧的操作。在方框图中 P 有可变延迟功能，这主要是运动估计运算时间不等，加之环形滤波器也有延迟，在前帧和后帧相减时及与预测帧相加时，均需对准时间，因此，需要可变延迟功能。综上所述，信源编码器有如下几种信号：帧内、帧间系数；帧内、帧间模式选择信号；帧内、帧间宏块量化步长；图像数据传输与否信号；帧间模式宏块运动矢量；环路滤波器是否使用指示信号。综观 H.261 标准建议，可以看出它给研制人员留有很大余地，可结合实用加以发挥。

（3）变换(DCT)

在该标准中采用性能较好的 DCT 变换。以 8×8 像块为基本单元，每像素 8 位量化 0～256 电平，再下移 128，亮度与色度信号一样处理。但色度信号已上移 128，故在此又恢复原样。原因是色度信号经常有正有负，而亮度信号均为正。平移后数据处在零电平附近，这样可以降低传输比特率。平移后的数据可进行 DCT 变换，即

$$F(u,v)=\frac{1}{4}C(u)C(v)\sum_{x}^{7}\sum_{y=0}^{7}f(x,y)\cdot\cos\frac{\pi(2X+1)u}{16}\cdot\cos\frac{\pi(2Y+1)v}{16}\tag{5-142}$$

$$u,v=0,1,2,\cdots,7$$

$$f(x,y)=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)F(u,v)\cdot\cos\frac{\pi(2X+1)u}{16}\cdot\cos\frac{\pi(2Y+1)v}{16}\tag{5-143}$$

$$x,y=0,1,2,\cdots,7$$

$$C(u)C(v)=\begin{cases}1/\sqrt{2} & u,v=0\\ 1 & u,v\neq 0\end{cases}$$

在空域中排列顺序为：左上角为像素在 x，y 坐标的原点，x 从左到右为 0～7，y 从上到下为 0～7。

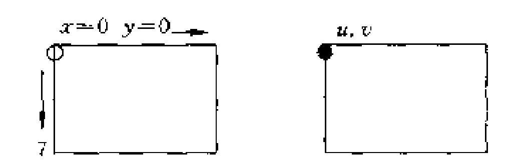


图 5-64 变换块的排列顺序

在频域 u,v 坐标系中，左上角为直流系数，右下角为 u 和 v 两个方向的最高空间频率系数。左上方系数的平方代表低频能量，右下方个系数值的平方代表图像的高频能量。

(4) 量化

H.261 标准的量化公式如下：

$$Q(u,v) = \text{取整数}\left(\frac{F(u,v)}{2q}S\right) \tag{5-144}$$

式中：DCT 系数 $F(u,v)$ 取绝对值，S 表示正负号，$S=0$ 表示正，$S=1$ 表示负。q 为量化步长，取 1～31，乘 2 后为 2～62。因为有很多 $F(u,v)$ 值较小，因此，$Q(u,v)$ 计算后有不少为零，从而使 bit 率降低。对每宏块 MB 有 6 个 8×8 块，量化步长都一样，q 是由缓冲器存储余量决定，余量大 q 取值低，输出 $Q(u,v)$ 值提高，同时 bit 率增加。若余量小，则 q 取值高，使输出 $Q(u,v)$ 值降低，产生许多零值，使传输 bit 率降低。量化中的取整过程，采取四舍五入的方法实现。

Q^{-1} 是反量化，由下式决定：

$$F'(u,v) = 2qQ(u,v) \tag{5-145}$$

该值是 $F(u,v)$ 的近似值。

5. 运动补偿

在 H.261 标准中，采用了运动补偿技术。在帧间编码中，需要传输前后帧宏块 MB 的差值，即：运动矢量 MV，其表达式为

$$\text{MV}(H,V) = \min_{h,v}\sum_{y=1}^{16}\sum_{x=1}^{16}|Y(x,y) - P(x+h,y+v)| \tag{5-146}$$

式中 min 表示搜索最小值。h,v 表示水平和垂直方向搜索像素数，对于前帧亮度信号最大搜索范围为 −15 像素到 +15 像素，后帧亮度信号的 MB 有 16×16 个像素，相应的色度块只有 8×8 个像素，所得到的运动矢量坐标除以 2。上式中的 MV(H,V) 除了表示所找到的最小差值外，其中 H、V 表示前帧中匹配宏块 MB 的位置。这些操作称为运动估计，运动估计的目的是找到运动矢量。通过运动估计在前帧中找到匹配宏块 MB 后，需要传输前后帧匹配宏块间像素差值矩阵，即：

$$\Delta\text{MB}_{xy} = Y(x,y) - P(x+H,y+V) \tag{5-148}$$

式中，ΔMB_{xy} 的 x 指水平方向的 16 个像素的差值，y 指垂直方向的 16 个像素的差值。如果原来像素灰度值为 −128 到 127，相减后成为 −255 到 255，除了亮度差值外，还有色度 U、V 的 64 个差值。按上述方法求得的宏块差值就是预测误差。每一宏块的预测误差再经过 DCT 变换、量化、编码后加以传送。由此可见，帧间编码对 P 帧编码的基本过程就是运动矢量的寻找，预测误差的计算，变换及编码。这一过程称之为运动补偿。运动补偿技术的运用，不仅可降低码率，而且可大大改善图像质量。

6. 环路滤波器

环路滤波器是一低通滤波器，其功能是消除高频噪声。通常，环路滤波器接在运动补偿环

路内，用以消除边缘赝像。环路滤波器的参数也不属于H. 261的标准，因此它可以根据需要而更改。

5.7.2 H. 261解码原理

H. 261的解码原理如图5-65所示。

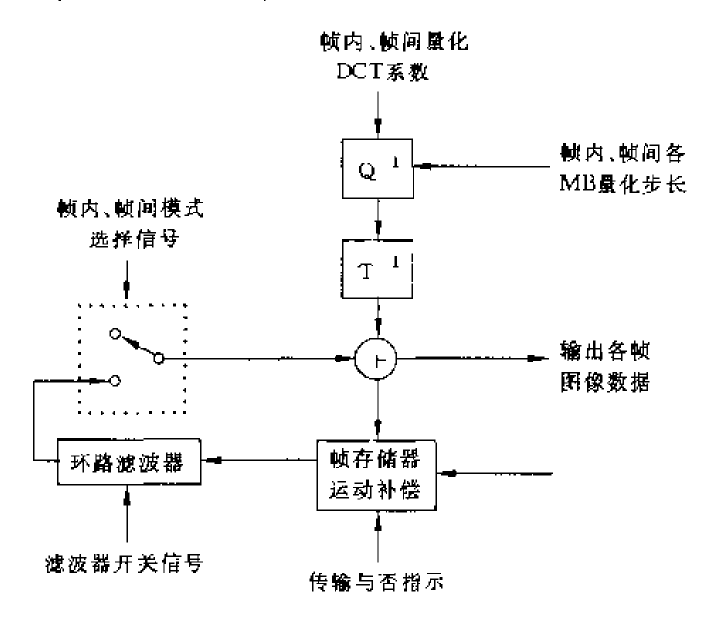


图5-65 解码方框图

该解码可分为帧内和帧间模式。

(1) 帧内解码模式

在帧内模式时，编码器传来的宏块MB量化DCT系数送入反量化器Q^{-1}，同时，送入该宏块的量化步长q，按照基本公式(5-144)可得到DCT系数$F'(u,v)$，经反DCT变换后得到像素值$f'(x,y)$。当帧内解码时，开关S处在上方，加法器分两路输出图像数据，一路供输出，另一路送入帧解码器，供帧间解码用。

(2) 帧间解码模式

在帧间模式下，编码器提供的DCT预测误差和量化步长送入反量化器Q^{-1}然后，再经DCT反变换得到预测误差$Y(x,y)-P(x+H,y+V)$，这时候，与MB相对应大运动矢量坐标H和V送入帧存储器，将前帧匹配信号从帧存储器取出送入环路滤波器，乘上滤波系数得到$P_Y(x+H,y+V)$。在帧间模式下，开关S倒向下方，则信号经开关送入加法器，与预测误差相加后得到$Y(x,y)$。加法器输出分两路，一路输出，另一路进入帧存储器，此时存储的是P帧备用，它是下一P帧的前帧。

5.7.3 H. 261的图像复用编码

为传输的需要，在CCITT建议的算法中，将CIF和QCUF划分成层次结构，即：图像层，块组层(GOB)，宏块层(MB)和块层(B)，共4个层次。

块组层(GOB)包含33个宏块(MB)，横向11块、纵向三行，如图5-66所示。

一个宏块(MB)由4个8×8亮度块(Y)，2个色度块(U,V)组成，因此，MB有6个数据块，如图5-67所示。数据块包含8×8个像素，这是亮度和色度信号的基本编码单元。

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

图 5-66 GOB 中宏块的顺序

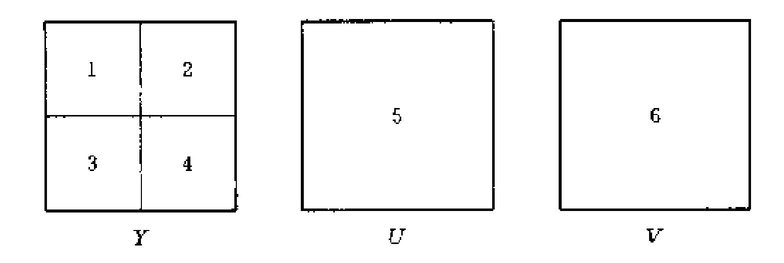


图 5-67 宏块 MB 数据顺序

按 CIF 格式的设计，一帧 CIF 图像有 12 个 GOB，等于 12×33 个宏块，又等于 936×6 数据块，共有 2376 个数据块(B)，其中 1584 个亮度块(Y)，396 个 U 块，396 个 V 块，共有 152064 个像素。一帧图像的 GOB 排列如图 5-68 所示：对于 CIF 格式有 12 个块组，一幅 QCIF 图像由 3 个块组组成。



图 5-68 GOB 的顺序

图像复用编码把以上层次的数据按一定的方式连接起来，构成一帧数据流。数据流的安排如图 5-69 所示。

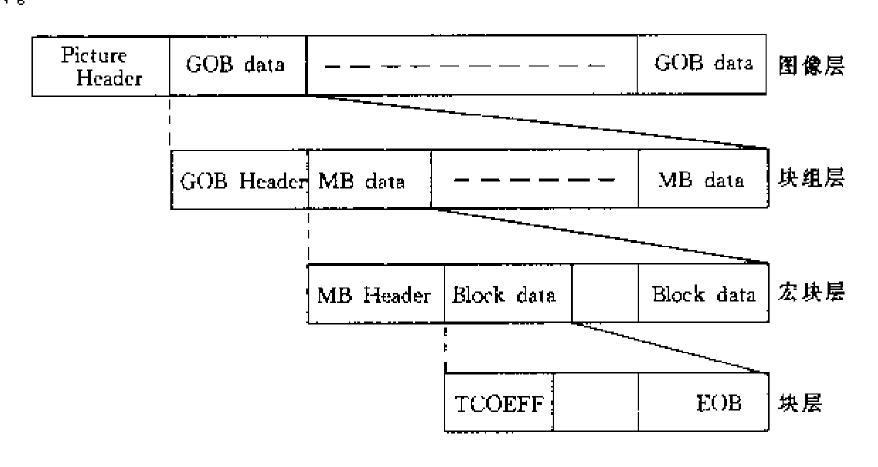


图 5-69 图像复用编码的数据安排

图像流由帧首和数据组成，帧首有20位的图像起始码、帧计数码、帧类型码等其他格式、时间参数(帧数)等信息；图像头块组层，由块组头和宏块组成。它包含16位GOB起始码、块组编号码、块组量化步长、备用插入信息码等及33块宏块；宏块层包含块首(宏块头)信息，即：变长地址码，宏块类型变长码，指明帧内帧间有无运动参数，有无循环滤波器，不需传的DCT系数等信息；最后是数据块B，它包括块头和数据块。变换系数遵循TCOEFF安排，每块8×8个数据，左上角为直流系数，其他是交流系数。由于视觉对低频信号较为敏感，并且高频系数零值较多，所以，采用之字形扫描方法，即图5-70所示。

在H.261中提供了TCOEFF编码表，帧间编码的第一个系数是直流系数。帧内编码中直流系数先用步长8量化，在用8位固定长度编码，关于帧内编码H.261标准也提供了编码表。数据块编码结束时，用结束符EOB标示。

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

图5-70　之字形扫描顺序

5.7.4　传输缓冲器与传输编码

在H.261编码标准中设置缓冲器的目的是为了协调编码输出的比特率和传输网的比特率。当编码输出的比特率过高时，缓冲器用于存储，同时令编码器提高量化步长，从而降低比特率，当编码器输出比特率过低时，缓冲器将控制编码器降低量化步长，从而提高比特率。所以，必须确定缓冲器的容量。这里HRD与编码器应有统一的时钟，相同的图像格式，缓冲器容量可由下式计算：

$$B = \frac{4R_{max}}{30} \tag{5-148}$$

其中R_{max}是最高图像传输比特率，分母中的30是帧频，如果采用PAL制式则选25。因此，$R_{max}=P\times 64$kb/s，如果$P=30$，则$R_{max}=2048$kb/s。编码器中的存储量与HRD一样，如图5-71所示的B＝256kb。

在H.261标准建议中还设计了纠错编码，该方案使用BCH码，这是一种循环冗余校验码，它是线性码的子集。码长为511位，其中信息码元493位，校验码元18位，如图5-72所示。

图5-72表示帧结构，第一行表示一个帧群，一个帧群有8帧，一帧有512位，每帧都有一个帧头，由$S_1S_2\cdots S_8$代表。帧头的码位规定为$S_1S_2S_3S_4S_5S_6S_7S_8=00011011$，帧头码位用作同步信号。在数据结构中，F表示填满指示，F取1表示其后边有492位数据，F取0，表示后边无数据，此时492位全部取1。18位校验码为：011011010100011011。当由缓冲存储器送来数据时，首先把这些数据分成每492位为一组，加一位F码成为493位，然后，该码送入BCH编

码器，经编码后成为 511 位，再加上同步位成为 512 位码。18 位校验位由下式形成：

$$g(x)=(x^9+x^4+1)(x^9+x^6+x^4+x^3+1)$$

及数据多项式

$$f(x)=v_0+v_1x+v_2x^2+\cdots+v_{492}x^{492}$$

其中 $v_i=0$ 或 1，$i=1,2,\cdots,493$。

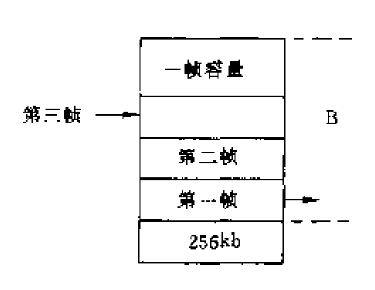


图 5-71　传输缓冲容量

这里 BCH 码的码长 n，信息码元 k 和纠错能力之间符合如下关系：

$$n=2^m-1 \qquad n-k\leqslant mt$$

式中，m 为大于 3 的正整数，由于 $n-k=18\leqslant 9t$，因此，$t=2$。它表明可纠 2 位误差。也就是说，如果出现 2 位码误差，BCH 码可加以纠正，如大于 2 位码误差就会出现误码。

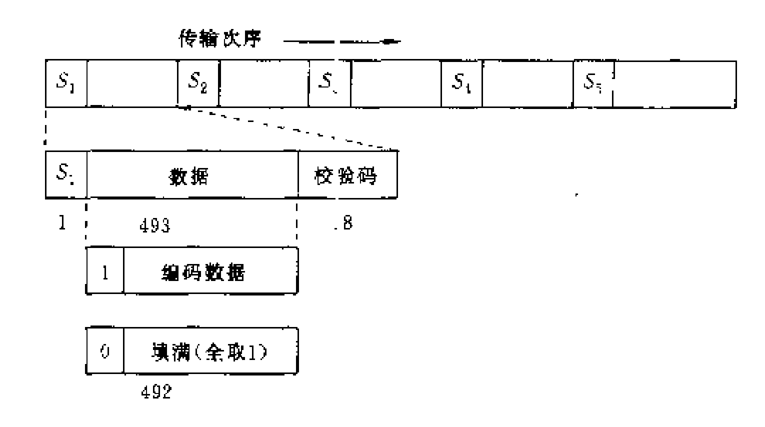


图 5-72　纠错帧安排

以上就是 CCITT H.261 编码标准的基本原理，若读者需要更详细的标准内容可参考标准手册。

思　考　题

1. 信源编码的目的是什么？信道编码的目的又是什么？

2. 按 Kunt 的提法，什么是第一代编码？第二代编码的特点是什么？

3. 图像编码的保真度准则是什么？

4. 试画出 PCM 图像编码的原理框图，如何提高 PCM 编码的量化信噪比？

5. 什么是非线性 PCM 编码？它有什么优点？

6. 试述两种压扩特性及其对小信号的信噪比性能的改善。

7. 试述编码效率和冗余度的概念及如何计算编码效率和冗余度。

8. 编码的基本要求是什么？试述其含义。

9. 何为匹配编码？

10. 有如下之信源 $\boldsymbol{X}$，

$$\boldsymbol{X}=\begin{Bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \\ P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \end{Bmatrix}$$

其中：$P_1=0.20$，$P_2=0.09$，$P_3=0.11$，$P_4=0.13$，$P_5=0.07$，$P_6=0.12$，$P_7=0.08$，$P_8=$

0.20。

① 试将该信源编为 Huffamn 码，并计算信源的熵、平均码长、编码效率及冗余度。

② 将该形源编成 Shannon-Fano 码并计算信源的熵、平均码长、编码效率和冗余度。

11. 试述预测编码的基本原理。

12. 试述 DM 编、解码的基本原理，画出方框图并分析其量化信噪比。

13. 试述 DPCM 的编码原理，画出原理方框图，分析其量化信噪比，同时，与 DM 做一比较。

14. 正交变换有哪些性质可供编码利用？

15. 试述 WHT 变换编码的原理框图。

16. 何为最佳变换，最佳的准则是什么？

17. 最佳变换的实用难点是什么？

18. 何为区域编码？何为门限编码？

19. 图像编码有哪些国际标准？其基本应用对象是什么？

20. 试述 H.261 的 CIF 和 QCIF 含义？

21. 试说明 H.261 中的 I、B、P 帧的含义。

22. 试述 H.261 的数据流安排。

第6章　图像复原

图像复原是图像处理的另一重要课题。它的主要目的是改善给定的图像质量。当给定了一幅退化了的或者受到噪声污染了的图像后，利用退化现象的某种先验知识来重建或恢复原有图像是图像复原处理的基本过程。可能的退化有光学系统中的衍射、传感器非线性畸变、光学系统的像差、摄影胶片的非线性、大气湍流的扰动效应、图像运动造成的模糊以及几何畸变等等。噪声干扰可以由电子成像系统传感器、信号传输过程或者胶片颗粒性造成。各种退化图像的复原都可归结为一种过程，具体地说就是把退化模型化，并且采用相反的过程进行处理，以便恢复出原图像。本章将主要讨论一些基本的复原技术。

6.1　退化模型

图像恢复处理的关键问题在于建立退化模型。在第2章已提到在用数学方法描述图像时，它的最普遍的数学表达式为$I=f(x,y,z,\lambda,t)$。这样一个表达式可以代表一幅活动的、彩色的立体电视图像。当研究的是静止的、单色的、平面的图像时，则其数学表达式就简化为$I=f(x,y)$。基于这样的数学表达式，可建立退化模型如图6-1所示的形式。由图6-1的模型可见，一幅纯净的图像$f(x,y)$是由于通过了一个系统H及加入外来加性噪声$n(x,y)$而使其退化为一幅图像$g(x,y)$的。

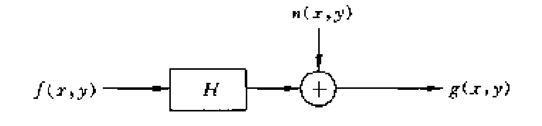


图6-1　图像退化模型

图像复原可以看成是一个估计过程。如果已经给出了退化图像$g(x,y)$并估计出系统参数H，从而可近似地恢复$f(x,y)$。这里，$n(x,y)$是一种统计性质的信息。当然，为了对处理结果作出某种最佳的估计，一般应首先明确一个质量标准。

6.1.1　系统H的基本定义

根据图像的退化模型及复原的基本过程可见，复原处理的关键在于对系统H的基本了解。就一般而言，系统是某些元件或部件以某种方式构造而成的整体。系统本身所具有的某些特性就构成了通过系统的输入信号与输出信号的某种联系。

系统的分类方法很多。例如，系统可分为线性系统和非线性系统；时变系统和非时变系统；集中参数系统和分布参数系统；连续系统和离散系统等等。

线性系统就是具有均匀性和相加性的系统。对于图6-1所示的系统来说，可表示成下式：

$$g(x,y)=H\cdot[f(x,y)]+n(x,y) \tag{6-1}$$

如果暂不考虑加性噪声$n(x,y)$的影响，而令$n(x,y)=0$时，则有

$$g(x,y)=H\cdot[f(x,y)] \tag{6-2}$$

如果输入信号为 $f_1(x,y)$，$f_2(x,y)$，对应的输出信号为 $g_1(x,y)$，$g_2(x,y)$，通过系统后有下式成立：

$$\begin{aligned}H\cdot[k_1f_1(x,y)]+k_2f_2(x,y)] &= H\cdot[k_1f_1(x,y)]+H\cdot[k_2f_2(x,y)]\\ &= k_1g_1(x,y)+k_2g_2(x,y)\end{aligned} \tag{6-3}$$

那么，系统 H 是一个线性系统。其中 k_1，k_2 为一常数。如果 $k_1=k_2=1$，则

$$\begin{aligned}H\cdot[f_1(x,y)+f_2(x,y)] &= H\cdot[f_1(x,y)]+H\cdot[f_2(x,y)]\\ &= g_1(x,y)+g_2(x,y)\end{aligned} \tag{6-4}$$

式(6-3)及式(6-4)说明，如果 H 为线性系统，那么，两个输入之和的响应等于两个响应之和。显然，线性系统的特性为求解多个激励情况下的输出响应带来很大方便。

如果一个系统的参数不随时间变化，即称为时不变系统或非时变系统。否则，就称该系统为时变系统。与此概念相对应，对于二维函数来说，如果

$$H\cdot[f(x-\alpha,y-\beta)]=g(x-\alpha,y-\beta) \tag{6-5}$$

则 H 是空间不变系统(或称为位置不变系统)，式中的 α 和 β 分别是空间位置的位移量。这说明了图像中任一点通过该系统的响应只取决于在该点的输入值，而与该点的位置无关。

由上述基本定义可见，如果系统 H 有式(6-3)和式(6-5)的关系，那么，系统就是线性的和空间位置不变的系统。在图像复原处理中，尽管非线性和空间变化的系统模型更具普遍性和准确性，但是，它却给处理工作带来巨大的困难，它常常没有解或者很难用计算机来处理。因此，在图像复原处理中，往往用线性和空间不变性的系统模型加以近似。这种近似的优点是使线性系统理论中的许多理论可直接用于解决图像复原问题，所以图像复原处理特别是数字图像复原处理主要采用线性的，空间不变的复原技术。

6.1.2 连续函数退化模型

在线性系统理论中，曾定义了单位冲激信号 $\delta(t)$。它是一个振幅在原点之外所有时刻为零，在原点处振幅为无限大、宽度无限小，面积为 1 的窄脉冲。其时域表达式为

$$\begin{cases}\int_{-\infty}^{+\infty}\delta(t)\mathrm{d}t=1 & t=0\\ \delta(t)=0 & t\neq 0\end{cases} \tag{6-6}$$

如果冲激信号 $\delta(t)$ 有一个 t_0 时刻的延迟，那么

$$\begin{cases}\int_{-\infty}^{+\infty}\delta(t-t_0)\mathrm{d}t=1 & t=t_0\\ \delta(t-t_0)=0 & t\neq t_0\end{cases} \tag{6-7}$$

冲激信号的一个重要特性是取样特性。由于 $\delta(t)$ 除了 $t=0$ 外其他值均为零，所以有

$$\int_{-\infty}^{+\infty}\delta(t)f(t)\mathrm{d}t=\int_{-\infty}^{+\infty}\delta(t)f(0)\mathrm{d}t=f(0)\int_{-\infty}^{+\infty}\delta(t)\mathrm{d}t=f(0) \tag{6-8}$$

同理，当 $t=t_0$ 时有

$$\int_{-\infty}^{+\infty}\delta(t-t_0)f(t)\mathrm{d}t=\int_{-\infty}^{+\infty}\delta(t-t_0)f(t_0)\mathrm{d}t=f(t_0)\int_{-\infty}^{+\infty}\delta(t-t_0)\mathrm{d}t=f(t_0) \tag{6-9}$$

冲激函数的另外一个取样公式就是卷积取样，即

$$\int_{-\infty}^{+\infty}f(x-t)\delta(t)\mathrm{d}t=f(x) \tag{6-10}$$

上述的一维时域冲激函数 $\delta(t)$ 不难推广到二维空间域中。如果推广至二维空间，那么可定义 $\delta(x,y)$ 为冲激函数。$\delta(x-\alpha,y-\beta)$ 就是有延迟的冲激函数。显然，可以把 $f(x,y)$ 写成下式形式：

$$f(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(\alpha,\beta)\delta(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-11}$$

根据 $g(x,y)=H\cdot[f(x,y)]+n(x,y)$ 的关系，如果令 $n(x,y)=0$，则有下式成立：

$$g(x,y)=H\cdot[f(x,y)]=H\cdot\left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(\alpha,\beta)\delta(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]$$

由于 H 是线性算子，所以

$$\begin{aligned}g(x,y)&=H\cdot[f(x,y)]\\&=H\cdot\left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(\alpha,\beta)\delta(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\\&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}H\cdot[f(\alpha,\beta)\delta(x-\alpha,y-\beta)]\mathrm{d}\alpha\mathrm{d}\beta\\&=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(\alpha,\beta)H\cdot\delta(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta\end{aligned} \tag{6-12}$$

令 $h(x,\alpha,y,\beta)=H\cdot\delta(x-\alpha,y-\beta)$

则

$$g(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(\alpha,\beta)h(x,\alpha,y,\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-13}$$

其中 $h(x,\alpha,y,\beta)$ 就是系统 H 的冲激响应。也就是说，$h(x,\alpha,y,\beta)$ 是系统 H 对坐标为 α,β 处的冲激函数 $\delta(x-\alpha,y-\beta)$ 的响应。在光学中，冲激为一光点，所以 $h(x,\alpha,y,\beta)$ 又称为点扩散函数(PSF)。

式(6-13)就是线性系统理论中非常重要的费雷德霍姆(Fredholm)积分。式(6-13)指出，如果系统 H 对冲激函数的响应为已知，则对任意输入 $f(\alpha,\beta)$ 的响应可用式(6-13)求得。换句话说，线性系统 H 完全可由其冲激响应来表征。

在空间位置不变的情况下，

$$H\cdot\delta(x-\alpha,y-\beta)=h(x-\alpha,y-\beta) \tag{6-14}$$

在这种情况下，显然

$$g(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(\alpha,\beta)h(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-15}$$

这说明，系统 H 加入输入信号的响应就是系统输入信号与冲激响应的卷积积分。

在有加性噪声的情况下，前述的线性退化模型可表示为

$$g(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(\alpha,\beta)h(x,\alpha,y,\beta)\mathrm{d}\alpha\mathrm{d}\beta+n(x,y) \tag{6-16}$$

当然，在上述情况中，都假设噪声与图像中的位置无关。

式(6-16)就是我们主要研究的连续函数的退化模型。

6.1.3 离散的退化模型

连续函数的退化模型是由输入函数 $f(\alpha,\beta)$ 和点扩散函数相乘后再积分来表示的。如果把 $f(\alpha,\beta)$ 和 $h(x-\alpha,y-\beta)$ 进行均匀取样后就可引伸出离散的退化模型。为了研究离散的退化模型，不妨用一维函数来说明基本概念，然后再推广至二维情况。

假设有两个函数 $f(x)$ 和 $h(x)$，它们被均匀取样后分别形成 A 维和 B 维的阵列。在这种情况下，$f(x)$ 变成在 $x=0,1,2,\cdots,A-1$ 范围内的离散变量，$h(x)$ 变成在 $x=0,1,2,\cdots,B-1$ 范围内的离散变量。由此，连续函数退化模型中的连续卷积关系就演变为离散卷积关系。

如果 $f(x)$ 和 $h(x)$，都是具有周期为 N 的序列，那么，它们的时域离散卷积可定义为下式之形式：

$$g(x) = \sum_{m} f(m)h(x-m) \tag{6-17}$$

显然，$g(x)$ 也是具有周期 N 的序列。周期卷积可用常规卷积法计算，也可用卷积定理进行快速卷积计算。

如果 $f(x)$ 和 $h(x)$ 均不具备周期性，则可以用延拓的方法使其成为周期函数。为了避免折叠现象，可以令周期 $M \geqslant A+B-1$，使 $f(x)$，$h(x)$ 分别延拓为下列离散阵列：

$$f_e(x) = \begin{cases} f(x) & 0 \leqslant x \leqslant A-1 \\ 0 & A-1 < x \leqslant M-1 \end{cases}$$

$$h_e(x) = \begin{cases} h(x) & 0 \leqslant x \leqslant B-1 \\ 0 & B-1 < x \leqslant M-1 \end{cases} \tag{6-18}$$

这样延拓后，可得到一个离散卷积退化模型：

$$g_e(x) = \sum_{m=0}^{M-1} f_e(m)h_e(x-m) \tag{6-19}$$

式中 $x=0,1,2,\cdots,M-1$。显然，$g_e(x)$ 的周期也是 M。经过这样的延拓处理，一个非周期的卷积问题就变成了周期卷积问题了。因此也就可以用快速卷积法进行运算了。

如果用矩阵来表示上述离散退化模型，可写成下式之形式：

$$[g] = [H][f] \tag{6-20}$$

其中

$$[f] = \begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ f_e(M-1) \end{bmatrix} \tag{6-21}$$

$$[g] = \begin{bmatrix} g_e(0) \\ g_e(1) \\ \vdots \\ g_e(M-1) \end{bmatrix} \tag{6-22}$$

$[H]$ 是 $M \times M$ 阶矩阵，

$$[H] = \begin{bmatrix} h_e(0) & h_e(-1) & h_e(-2) & \cdots & h_e(-M+1) \\ h_e(1) & h_e(0) & h_e(-1) & \cdots & h_e(-M+2) \\ h_e(2) & h_e(1) & h_e(0) & \cdots & h_e(-M+3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_e(M-1) & h_e(M-2) & h_e(M-3) & \cdots & h_e(0) \end{bmatrix} \tag{6-23}$$

由于 $h_e(x)$ 具有周期性，所以 $h_e(x)=h_e(M-x)$，利用这一性质，式(6-23)又可以写成下式形式：

$$[H]=\begin{bmatrix} h_e(0) & h_e(M-1) & h_e(M-2) & \cdots & h_e(1) \\ h_e(1) & h_e(0) & h_e(M-1) & \cdots & h_e(2) \\ h_e(2) & h_e(1) & h_e(0) & \cdots & h_e(3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_e(M-1) & h_e(M-2) & h_e(M-3) & \cdots & h_e(0) \end{bmatrix} \tag{6-24}$$

由于 $h_e(x)$ 的周期性，$[H]$ 为一个循环矩阵。

上述基本模型不难推广至二维情况。如果给出 $A\times B$ 大小的数字图像以及 $C\times D$ 大小的点扩散函数，可首先作成大小为 $M\times N$ 的周期延拓图像，即：

$$f_e(x,y)=\begin{cases} f(x,y) & 0\leqslant x\leqslant A-1 \\ & 0\leqslant y\leqslant B-1 \\ & A< x\leqslant M-1 \\ 0 & B< y\leqslant N-1 \end{cases} \tag{6-25}$$

$$h_e(x,y)=\begin{cases} h(x,y) & 0\leqslant x\leqslant C-1 \\ & 0\leqslant y\leqslant D-1 \\ & C< x\leqslant M-1 \\ 0 & D< y\leqslant N-1 \end{cases} \tag{6-26}$$

这样延拓后，$f_e(x,y)$ 和 $h_e(x,y)$ 分别成为二维周期函数。它们在 x 和 y 方向上的周期分别为 M 和 N。由此得到二维退化模型为一个二维卷积形式：

$$g_e(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f_e(m,n)h_e(x-m,y-n) \tag{6-27}$$

式中：$x=0,1,2,\cdots,M-1$；$y=0,1,2,\cdots,N-1$，卷积函数 $g(x,y)$ 也为周期函数，其周期与 $f_e(x,y)$ 和 $h_e(x,y)$ 一样。为避免重叠，同样要按下式规则延拓：

$$M\geqslant A+C-1 \qquad N\geqslant B+D-1 \tag{6-28}$$

式(6-27)的模型同样可用矩阵来表示：

$$[g]=[H][f] \tag{6-29}$$

其中 $[g]$，$[f]$ 代表 MN 维列向量。这些列向量是由 $M\times N$ 维的函数矩阵 $[f_e(x,y)]$，$[g_e(x,y)]$ 的各行堆积而成的。例如，$[f]$ 的第一组 N 个元素是 $[f_e(x,y)]$ 的第一行元素，第二组 N 个元素是由 $[f_e(x,y)]$ 的第二行元素得到的等等。因此，式(6-29)中的 $[g]$ 和 $[f]$ 是 MN 维向量矩阵，即 $[g]$，$[f]$ 为 $(MN)\times 1$ 维矩阵。而 $[H]$ 为 $MN\times MN$ 维矩阵，即

$$[H]=\begin{bmatrix} [H_0] & [H_{M-1}] & [H_{M-2}] & \cdots & [H_1] \\ [H_1] & [H_0] & [H_{M-1}] & \cdots & [H_2] \\ [H_2] & [H_1] & [H_0] & \cdots & [H_3] \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ [H_{M-1}] & [H_{M-2}] & [H_{M-3}] & \cdots & [H_0] \end{bmatrix} \tag{6-30}$$

每个部分 $[H_j]$ 是由延拓函数 $h_e(x,y)$ 的 j 行构成的，构成方法如下式：

$$[H_j]=\begin{bmatrix} h_e(j,0) & h_e(j,N-1) & h_e(j,N-2) & \cdots & h_e(j,1) \\ h_e(j,1) & h_e(j,0) & h_e(j,N-1) & \cdots & h_e(j,2) \\ h_e(j,2) & h_e(j,1) & h_e(j,0) & \cdots & h_e(j,3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_e(j,N-1) & h_e(j,N-2) & h_e(j,N-3) & \cdots & h_e(j,0) \end{bmatrix} \tag{6-31}$$

这里$[H_j]$是一个循环矩阵，$[H]$的分块$[H_j]$的下标也是循环方式标注。因此，$[H]$是一个分块循环矩阵。

一个更加完善的退化模型应加上噪声项。所以离散退化模型的完整形式如下式所示：

$$g_e(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f_e(m,n)h_e(x-m,y-n)+n_e(x,y) \tag{6-32}$$

其矩阵形式如下：

$$[g]=[H][f]+[n] \tag{6-33}$$

式中$[n]$也是MN维列向量。

上述离散退化模型都是在线性的空间不变的前提下推出的。目的是在给定了$g(x,y)$，并且知道$h(x,y)$和$n(x,y)$的情况下，估计出理想的原始图像$f(x,y)$。但是，要想从式(6-33)得到$f(x,y)$，对于实用大小的图像来说，处理工作是十分艰巨的。例如，对于一般精度的图像来说，$M=N=512$，此时$[H]$的大小为$MN\times MN=(512)^2\times(512)^2=262144\times262144$。因此，要直接得到$[f]$则需要求解262144个联立方程组。其计算量之浩大是不难想像的。为了解决这样的问题，必须研究一些简化算法，由于$[H]$的循环性质，使得简化运算得以实现。

6.2 复原的代数方法

图像复原的主要目的是当给定退化的图像$\boldsymbol{g}$及$\boldsymbol{H}$和$\boldsymbol{n}$的某种了解或假设，估计出原始图像$\boldsymbol{f}$($\boldsymbol{g}=[g]$，$\boldsymbol{H}=[H]$，$\boldsymbol{n}=[n]$)。如果退化模型就是式(6-33)的形式，就可以用线性代数中的理论解决图像复原问题。

代数复原方法的中心是寻找一个估计$\hat{\boldsymbol{f}}$，它使事先确定的某种优度准则为最小。

6.2.1 非约束复原方法

由式(6-33)的退化模型可知，其噪声项为

$$\boldsymbol{n}=\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f} \tag{6-34}$$

在并不了解$\boldsymbol{n}$的情况下，希望找到一个$\hat{\boldsymbol{f}}$，使得$\boldsymbol{H}\hat{\boldsymbol{f}}$在最小二乘方意义上来说近似于$\boldsymbol{g}$。也就是说，希望找到一个$\hat{\boldsymbol{f}}$，使得

$$\|\boldsymbol{n}\|^2=\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2 \tag{6-35}$$

为最小。由定义可知：

$$\|\boldsymbol{n}\|^2=\boldsymbol{n}^{\mathrm{T}}\cdot\boldsymbol{n} \tag{6-36}$$

$$\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2=(\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}})^{\mathrm{T}}(\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}) \tag{6-37}$$

求$\|\boldsymbol{n}\|^2$最小等效于求$\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2$最小，即

$$\boldsymbol{J}(\hat{\boldsymbol{f}})=\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2 \tag{6-38}$$

实际上是求$\boldsymbol{J}(\hat{\boldsymbol{f}})$的极小值问题，这里选择$\hat{\boldsymbol{f}}$除了要求$\boldsymbol{J}(\hat{\boldsymbol{f}})$为最小外，不受任何其他条件约束，因此称为非约束复原。求(6-38)式的极小值方法就是用一般的求极值的方法。把$\boldsymbol{J}(\hat{\boldsymbol{f}})$对$\hat{\boldsymbol{f}}$微分，并使结果为零，即

$$\frac{\partial\boldsymbol{J}(\hat{\boldsymbol{f}})}{\partial\hat{\boldsymbol{f}}}=-2\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}})=0 \tag{6-39}$$

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\hat{\boldsymbol{f}}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g}$$

$$\hat{\boldsymbol{f}}=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \tag{6-40}$$

令 $M=N$，因此，$\boldsymbol{H}$ 为一方阵，并且设 $\boldsymbol{H}^{-1}$存在，则可求得 $\hat{f}$，

$$\hat{\boldsymbol{f}} = \boldsymbol{H}^{-1}(\boldsymbol{H}^{\mathrm{T}})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} = \boldsymbol{H}^{-1}\boldsymbol{g} \tag{6-41}$$

6.2.2 约束复原法

在最小二乘方复原处理中，为了在数学上更容易处理，常常附加某种约束条件。例如，可以令 $\boldsymbol{Q}$ 为 $\boldsymbol{f}$ 的线性算子，那么，最小二乘方复原问题可看成是使形式为 $\|\boldsymbol{Q}\hat{\boldsymbol{f}}\|^2$的函数，服从约束条件 $\|\boldsymbol{g}-\boldsymbol{H}\hat{\boldsymbol{f}}\|^2=\|\boldsymbol{n}\|^2$ 的最小化问题。而这种有附加条件的极值问题可用拉格朗日乘数法来处理。其处理方法如下：

寻找一个 $\hat{f}$，使下述准则函数为最小，

$$\boldsymbol{J}(\hat{\boldsymbol{f}}) = \|\boldsymbol{Q}\hat{\boldsymbol{f}}\|^2 + \lambda(\|\boldsymbol{g} - \boldsymbol{H}\hat{\boldsymbol{f}}\|^2 - \|\boldsymbol{n}\|^2) \tag{6-42}$$

式中 λ 为一常数，是拉格朗日系数。加上约束条件后，就可以按一般求极小值的方法进行求解。将式(6-42)对 $\hat{f}$ 微分，并使结果为零，则有

$$\frac{\partial \boldsymbol{J}(\hat{\boldsymbol{f}})}{\partial \hat{\boldsymbol{f}}} = 2\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\hat{\boldsymbol{f}} - 2\lambda\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{g} - \boldsymbol{H}\hat{\boldsymbol{f}}) = 0 \tag{6-43}$$

求解 $\hat{f}$，

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\hat{\boldsymbol{f}} + \lambda\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\hat{\boldsymbol{f}} - \lambda\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} = 0$$

$$\frac{1}{\lambda}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\hat{\boldsymbol{f}} + \lambda\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\hat{\boldsymbol{f}} = \lambda\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g}$$

$$\hat{\boldsymbol{f}} = \left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} + \frac{1}{\lambda}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \tag{6-44}$$

式(6-41)及式(6-44)是代数复原方法的基础。

6.3 逆滤波

6.3.1 逆滤波的基本原理

逆滤波复原法也叫做反向滤波法，基本原理如下。

如果退化图像为 $g(x,y)$，原始图像为 $f(x,y)$，在不考虑噪声的情况下，其退化模型用下式表示：

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)h(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-45}$$

这显然是一卷积表达式。由傅里叶变换的卷积定理可知有下式成立：

$$G(u,v) = H(u,v)\cdot F(u,v) \tag{6-46}$$

式中，$G(u,v)$，$H(u,v)$，$F(u,v)$分别是退化图像 $g(x,y)$，点扩散函数 $h(x,y)$，原始图像 $f(x,y)$的傅里叶变换。由式(6-46)，可得：

$$F(u,v) = \frac{G(u,v)}{H(u,v)} \tag{6-47}$$

$$f(x,y) = \mathscr{F}^{-1}[F(u,v)] = \mathscr{F}^{-1}\left[\frac{G(u,v)}{H(u,v)}\right] \tag{6-48}$$

这意味着，如果已知退化图像的傅里叶变换和“滤波”传递函数，则可以求得原始图像的傅里叶变换，经反傅里叶变换就可求得原始图像 $f(x,y)$。这里，$G(u,v)$除以 $H(u,v)$起到了反向滤波

的作用。这就是逆滤波法复原的基本原理。

在有噪声的情况下，逆滤波原理可写成如下形式：

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \tag{6-49}$$

$$F(u,v) = \frac{G(u,v)}{H(u,v)} - \frac{N(u,v)}{H(u,v)} \tag{6-50}$$

式中 $N(u,v)$是噪声 $n(x,y)$的傅里叶变换。

利用式(6-47)和式(6-50)进行复原处理时可能会发生下列情况：即在 u,v 平面上有些点或区域会产生 $H(u,v)=0$ 或 $H(u,v)$非常小的情况，在这种情况下，即使没有噪声，也无法精确地恢复 $f(x,y)$。另外，在有噪声存在时，在 $H(u,v)$的邻域内，$H(u,v)$的值可能比 $N(u,v)$的值小的多，因此由式(6-50)得到的噪声项可能会非常大，这样也会使 $f(x,y)$不能正确恢复。

一般来说，逆滤波法不能正确地估计 $H(u,v)$的零点，因此必须采用一个折衷的方法加以解决。实际上，逆滤波不是用 $1/H(u,v)$，而是采用另外一个关于 u,v 的函数 $M(u,v)$。它的处理框图如图 6-2 所示。

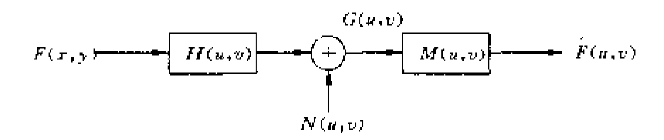


图 6-2　实际的逆滤波处理框图

在没有零点并且也不存在噪声的情况下，

$$M(u,v) = \frac{1}{H(u,v)}$$

图 6-2 的模型包括了退化和恢复运算。退化和恢复总的传递函数可用 $H(u,v)M(u,v)$来表示。此时有

$$\hat{F}(u,v) = [H(u,v)M(u,v)]F(u,v) \tag{6-51}$$

式中 $\hat{f}(x,y)$是 $f(x,y)$的估计值，$\hat{F}(u,v)$是 $\hat{f}(x,y)$的傅里叶变换。$H(u,v)$叫做输入传递函数，$M(u,v)$叫做处理传递函数，$H(u,v)M(u,v)$叫做输出传递函数。

一般情况下，$H(u,v)$的幅度随着离 u,v 平面原点的距离的增加而迅速下降，而噪声项 $N(u,v)$的幅度变化是比较平缓的。在远离 u,v 平面的原点时 $N(u,v)/H(u,v)$的值就会变得很大，而对于大多数图像来说 $F(u,v)$却变小，在这种情况下，噪声反而占优势，自然无法满意地恢复出原始图像。这一规律说明，应用逆滤波时仅在原点邻域内采用 $1/H(u,v)$方能奏效。换句话说，应使 $M(u,v)$在下述范围内选择：

$$M(u,v) = \begin{cases} \dfrac{1}{H(u,v)} & u^2 + v^2 \leqslant \omega_0^2 \\ 1 & u^2 + v^2 > \omega_0^2 \end{cases} \tag{6-52}$$

ω_0 的选择应该将 $H(u,v)$的零点排除在此邻域之外。

6.3.2　去除由均匀直线运动引起的模糊

1. 模糊模型

在获取图像时，由于景物和摄像机之间的相对运动，往往造成图像的模糊。其中由均匀直线运动所造成的模糊图像的恢复问题更具有一般性和普遍意义。因为变速的、非直线的运动在

某些条件下可以看成是均匀的、直线运动的合成结果。

假设图像 $f(x,y)$ 有一个平面运动，令 $x_0(t)$ 和 $y_0(t)$ 分别为在 x 和 y 方向上运动的变化分量。t 表示运动的时间。记录介质的总曝光量是在快门打开到关闭这段时间的积分。则模糊后的图像为

$$g(x,y) = \int_0^T f[x - x_0(t), y - y_0(t)]\mathrm{d}t \tag{6-53}$$

式中 $g(x,y)$ 为模糊后的图像。式(6-53)就是由目标物或摄像机相对运动造成图像模糊的模型。

2. 图像的恢复

令 $G(u,v)$ 为模糊图像 $g(x,y)$ 的傅里叶变换，对式(6-53)两边取傅里叶变换，得

$$\begin{aligned} G(u,v) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)\exp[-\mathrm{j}2\pi(ux+vy)]\mathrm{d}x\mathrm{d}y \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left\{\int_0^T f[x - x_0(t), y - y_0(t)]\mathrm{d}t\right\}\cdot\exp[-\mathrm{j}2\pi(ux+vy)]\mathrm{d}x\mathrm{d}y \end{aligned} \tag{6-54}$$

变换式(6-54)的积分次序，则有

$$G(u,v) = \int_0^T\left\{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f[x - x_0(t), y - y_0(t)]\cdot\exp[-\mathrm{j}2\pi(ux+vy)]\mathrm{d}x\mathrm{d}y\right\}\mathrm{d}t \tag{6-55}$$

由傅里叶变换的移位性质得到：

$$\begin{aligned} G(u,v) &= \int_0^T F(u,v)\exp\left\{-\mathrm{j}2\pi[ux_0(t) + vy_0(t)]\right\}\mathrm{d}t \\ &= F(u,v)\int_0^T \exp\left\{2 - \mathrm{j}2\pi[ux_0(t) + vy_0(t)]\right\}\mathrm{d}t \end{aligned} \tag{6-56}$$

如果令

$$H(u,v) = \int_0^T \exp\left\{-\mathrm{j}2\pi[ux_0(t) + vy_0(t)]\right\}\mathrm{d}t \tag{6-57}$$

则可得到下式：

$$G(u,v) = H(u,v)F(u,v) \tag{6-58}$$

这是已知的退化模型的傅里叶变换式。若 $x(t)$，$y(t)$ 的性质已知，传递函数可直接由式(6-57)求出，因此，$f(x,y)$ 可以恢复出来。

如果模糊图像是由景物在 x 方向上作均匀直线运动造成的，则模糊后图像任意点的值为

$$g(x,y) = \int_0^T f[x - x_0(t), y]\mathrm{d}t \tag{6-59}$$

式中 $x_0(t)$ 是景物在 x 方向上的运动分量。若图像总的位移量为 a，总的运动时间为 T，则运动的速率为 $x_0(t) = \frac{at}{T}$。由于只考虑在 x 方向的运动，所以 $y_0(t) = 0$。于是式(6-57)变为下式：

$$\begin{aligned} H(u,v) &= \int_0^T \exp[-\mathrm{j}2\pi u x_0(t)]\mathrm{d}t = \int_0^T \exp[-\mathrm{j}2\pi u\frac{at}{T}]\mathrm{d}t \\ &= \frac{T}{\pi ua}\sin(\pi ua)\mathrm{e}^{-\mathrm{j}\pi ua} \end{aligned} \tag{6-60}$$

由式(6-60)可见，当 $u = \frac{n}{a}$（n 为整数）时 $H(u,v) = 0$。在这些点上无法用逆滤波法恢复原图像。

当在区间 $0 \leqslant x \leqslant L$ 之外，$f(x,y)$ 为零或已知时，有可能避免式(6-60)引出的问题，并且可以根据在这一区间内对 $g(x,y)$ 的了解重建图像。

当只考虑 x 方向时，y 是时不变的，所以可以暂时忽略掉 y，式(6-53)可写做下式：

$$g(x)=\int_0^T f[x-x_0(t)]\mathrm{d}t=\int_0^T f\left(x-\frac{at}{T}\right)\mathrm{d}t \qquad 0\leqslant x\leqslant L \tag{6-61}$$

令 $\tau=x-\frac{at}{T}$，代入式(6-61)有

$$g(x)=\int_{x-a}^{x} f(\tau)\mathrm{d}\tau \qquad 0\leqslant x\leqslant L \tag{6-62}$$

对上式微分有

$$g'(x)=f(x)-f(x-a) \qquad 0\leqslant x\leqslant L$$

$$f(x)=g'(x)+f(x-a) \tag{6-63}$$

下面设置 n 个中间变量，以便推出一种递推解法。

设 $L=Ka$，K 为一整数，L 是 x 的取值范围，a 是图像内景物移动的总距离。则变量 x 可表示为下式：

$$x=z+ma \tag{6-64}$$

z 的取值在[0,a]之间，m 是 $\frac{x}{a}$ 的整数部分。显然，当 $x=L$ 时，有 $z=a$，$m=K-1$。将式(6-64)代入式(6-63)，则得

$$f(z+ma)=g'(z+ma)+f[z+(m-1)a] \tag{6-65}$$

设 $\phi(z)$ 为曝光期间在 $0\leqslant z<a$ 范围内移动的景物部分，即

$$\phi(z)=f(z-a) \qquad 0\leqslant z<a \tag{6-66}$$

通过 $\phi(z)$，用递推解法求解式(6-65)。

当 $m=0$ 时，

$$f(z)=g'(z)+f(z-a)=g'(z)+\phi(z)$$

当 $m=1$ 时，

$$f(z+a)=g'(z+a)+f(z)=g'(z+a)+g'(z)+\phi(z)$$

当 $m=2$ 时，

$$\begin{aligned}f(z+2a)&=g'(z+2a)+f(z+a)\\&=g'(z+2a)+g'(z+a)+g'(z)+\phi(z)\end{aligned}$$

以此类推，如果继续这一过程，将得到如下结果：

$$f(z+ma)=\sum_{k=0}^{m}g'(z+ka)+\phi(z) \tag{6-67}$$

由于 $x=z+ma$，因此，式(6-67)可表示为

$$f(x)=\sum_{k=0}^{m}g(x-ka)+\phi(x-ma) \qquad 0\leqslant x\leqslant Z \tag{6-68}$$

对于式(6-68)来说，$g(x)$是已知的劣化图像，要求得 $f(x)$则只需估计出 $\phi(x)$。

直接由模糊图像估算 $\phi(x)$的方法可如下进行。当 x 从 0 变到 Z，m 取 0,1,2,…,$K-1$ 的整数。ϕ 的自变量为$(x-ma)$，此变量总是在 $0\leqslant x-ma<a$ 范围内变化。要计算 $f(x)$值，而 x 值从 0 变到 L，m 将取 K 个值，所以 ϕ 将重复 K 次。

令

$$\tilde{f}(x)=\sum_{k=0}^{m}g'(x-ka) \tag{6-69}$$

则

$$\phi(x - ma) = f(x) - \hat{f}(x) \tag{6-70}$$

做一次变量置换，则得

$$\phi(x) = f(x + ma) - \hat{f}(x + ma) \tag{6-71}$$

如果在 $ma \leqslant x < (m+1)a$ 时，对上式两边进行计算，并且把 $m=0,1,2,\cdots,K-1$ 时的结果加起来，则

$$\sum_{m=0}^{K-1}\phi(x) = \sum_{m=0}^{K-1}f(x + ma) - \sum_{m=0}^{K-1}\hat{f}(x + ma) \tag{6-72}$$

由式(6-72)可见，左边当 m 从 0 变到 $K-1$ 时，$\phi(x)$均取相同的值($\phi(x)$与 m 无关)，所以在求和过程中只是 $\phi(x)$重复了 K 次，因此，

$$K\phi(x) = \sum_{m=0}^{K-1}f(x + ma) - \sum_{m=0}^{K-1}\hat{f}(x + ma) \tag{6-73}$$

将上式中的 m 换成 k，则

$$K\phi(x) = \sum_{k=0}^{K-1}f(x + ka) - \sum_{k=0}^{K-1}\hat{f}(x + ka) \tag{6-74}$$

两边同除以 K，则

$$\phi(x) = \frac{1}{K}\sum_{k=0}^{K-1}f(x + ka) - \frac{1}{K}\sum_{k=0}^{K-1}\hat{f}(x + ka) \tag{6-75}$$

式(6-75)中的第一项虽然是未知的，但是当 K 很大时，它趋于 $f(x)$的平均值，因此可以把第一项求和式看作一个常量 A，所以上式可近似于式(6-76)，即

$$\phi(x) \approx A - \frac{1}{K}\sum_{k=0}^{K-1}\hat{f}(x + ka) \qquad 0 \leqslant x < a \tag{6-76}$$

或者

$$\phi(x - ma) \approx A - \frac{1}{K}\sum_{k=0}^{K-1}\hat{f}[x + (k - m)a] \qquad 0 \leqslant x < L \tag{6-77}$$

又因为

$$\hat{f}(x) = \sum_{k=0}^{m}g'(x - ka)$$

代入式(6-77)，有

$$\begin{aligned}\phi(x - ma) &\approx A - \frac{1}{K}\sum_{m=0}^{K-1}\sum_{k=0}^{m}g'(x - ma) \\ &\approx A - \frac{1}{K}K \cdot mg'(x - ma) \approx A - mg'(x - ma)\end{aligned} \tag{6-78}$$

最后得到

$$f(x) \approx A - mg'(x - ma) + \sum_{k=0}^{m}g'(x - ka) \qquad 0 \leqslant x \leqslant L \tag{6-79}$$

再引入去掉了的变量 y，则

$$f(x,y) \approx A - mg'[(x - ma),y] + \sum_{k=0}^{m}g'[(x - ka)\quad y] \qquad 0 \leqslant x,\ y \leqslant L \tag{6-80}$$

这就是去除由 x 方向上均匀运动造成的图像模糊的表达式。

由上面的讨论可总结于下：

1）由水平方向均匀直线运动造成的图像模糊的模型及恢复的近似公式用以下两式表示：

$$g(x,y) = \int_{0}^{T} f\left[\left(x - \frac{at}{T}\right), y\right]\mathrm{d}t \tag{6-81}$$

$$f(x,y) \approx A - mg'[(x-ma),y] + \sum_{k=0}^{m} g'[(x-ka),y] \qquad 0 \leqslant x \quad y \leqslant L \tag{6-82}$$

式中 a 为总位移量，T 为总运动时间。

在计算机处理中，多用离散形式的公式，所以式(6-81)及式(6-82)的离散公式如下：

$$g(x,y) = \sum_{t=0}^{T-1} f\left[x - \frac{at}{T}, y\right] \cdot \Delta x \tag{6-83}$$

$$f(x,y) \approx A - m\left\{\left[g[(x-ma),y] - g[(x-ma-1),y]\right] \Big/ \Delta x\right\}$$
$$+ \sum_{k=0}^{m}\left\{\left[g[(x-ka),y] - g[(x-ka-1),y]\right] \Big/ \Delta x\right\} \quad 0 \leqslant x \quad y \leqslant L \tag{6-84}$$

2) 由垂直方向均匀直线运动造成的图像模糊模型及恢复的近公式用以下二式表示：

$$g(x,y) = \sum_{t=0}^{T-1} f\left(x, y - \frac{bt}{T}\right) \cdot \Delta y \tag{6-85}$$

$$f(x,y) \approx A - m\left\{\left[g[x,(y-mb)] - g[x,(y-mb-1)]\right] \Big/ \Delta y\right\}$$
$$+ \sum_{k=0}^{m}\left\{\left[g[x,(y-kb)] - g[x,(y-kb-1)]\right] \Big/ \Delta y\right\} \tag{6-86}$$

上述模糊图像的恢复处理结果如图 6-3 所示，(a) 是原始图像，(b)是模糊图像，(c) 是恢复后的图像　该处理的程序列在附录 7 中，可作一参考。



(a)



(b)



(c)

图 6-3　模糊图像及其复原处理结果

6.4 最小二乘方滤波

最小二乘方滤波也就是维纳滤波。它是使原始图像 $f(x,y)$ 及其恢复图像 $\hat{f}(x,y)$ 之间的均方误差最小的复原方法。

6.4.1 最小二乘方滤波的原理

设原始图像、相应的退化图像和噪声分别为 $f(x,y)$，$g(x,y)$ 和 $n(x,y)$。显然，它们有如下之关系式成立：

$$g(x,y)=\iint h(x-\alpha,y-\beta)f(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta+n(x,y) \tag{6-87}$$

式中 $g(x,y)$、$f(x,y)$ 和 $n(x,y)$ 分别为随机像场。式中噪声随机像场是不能精确知道的，但假定它的统计特性是已知的。因此，在给定了 $g(x,y)$ 时，仍然不能精确地求解 $f(x,y)$。在此，只能找出 $f(x,y)$ 的一个估计值 $\hat{f}(x,y)$，使得均方误差式：

$$e^2=E\left\{[f(x,y)-\hat{f}(x,y)]^2\right\} \tag{6-88}$$

最小，其中 $\hat{f}(x,y)$ 就叫给定 $g(x,y)$ 时 $f(x,y)$ 的最小二乘方估计。

为了便于数学处理，假定 $\hat{f}(x,y)$ 是 $g(x,y)$ 灰度级的线性函数，那么，

$$\hat{f}(x,y)=\iint m(x,y,\alpha,\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-89}$$

这里 $m(x,y,\alpha,\beta)$ 是在计算 (x,y) 处的 $\hat{f}(x,y)$ 时给予退化图像在 (α,β) 点的灰度级的权重。如果随机像场是均匀的，则加权函数只与 $(x-\alpha,y-\beta)$ 有关，所以

$$\hat{f}(x,y)=\iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-90}$$

将式(6-90)代入式(6-88)，则

$$e^2=E\left\{\left[f(x,y)-\iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]^2\right\} \tag{6-91}$$

显然，需要寻求使 e^2 最小的点扩散函数 $m(x,y)$。

可以证明，对于 xy 平面上所有的位置向量 (x,y) 和 (α',β') 都满足下式：

$$E\left\{\left[f(x,y)-\iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\times g(\alpha',\beta')\right\}=0 \tag{6-92}$$

的函数将使式(6-91)最小。

设 $m(x,y)$ 是一个满足式(6-92)的函数。任选一个其他函数 $m'(x,y)$，其均方误差由下式表示：

$$e'^2=E\left\{\left[f(x,y)-\iint m'(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]^2\right\} \tag{6-93}$$

现在可证明 $m'(x,y)=m(x,y)$ 时，式(6-93)最小。将式(6-93)改写于下：

$$\begin{aligned}e'^2=&E\left\{\left[f(x,y)-\iint m'(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]^2\right\}\\=&E\left\{\left[f(x,y)-\iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\right.\\&\left.+\iint[m(x-\alpha,y-\beta)-m'(x-\alpha,y-\beta)]g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right\}^2\end{aligned}$$

$$= E\left\{\left[f(x,y) - \iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]^2\right\}$$

$$+ E\left\{\iint[m(x-\alpha,y-\beta) - m'(x-\alpha,y-\beta)]g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right\}^2$$

$$+ 2E\left\{\left[f(x,y) - \iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\right.$$

$$\left.\times \left[\iint[m(x-\alpha,y-\beta) - m'(x-\alpha,y-\beta)]g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\right\} \tag{6-94}$$

由式(6-94)可见，第一项就是 e^2，第二项总是大于零的项，所以可写成：

$$e'^2 = e^2 + 正数 + 2E\left\{\left[f(x,y) - \iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\right.$$

$$\left.\times \left[\iint[m(x-\alpha,y-\beta) - m'(x-\alpha,y-\beta)]g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\right\} \tag{6-95}$$

式(6-95)是两项之积，且都包含有(α,β)的积分，把后一个积分变量改为(α',β')并互换积分与求期望的次序，则有

$$e'^2 = e^2 + 正数 + \iint E\left\{\left[f(x,y) - \iint m(x-\alpha,y-\beta)g(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta\right]\right.$$

$$\left.\times g(\alpha',\beta')\right\}[m(x-\alpha',y-\beta') - m'(x-\alpha',y-\beta')]\mathrm{d}\alpha'\mathrm{d}\beta' \tag{6-96}$$

显然，式(6-96)中第三项满足式(6-92)，因而第三项为零，式(6-96)变为

$$e'^2 = e^2 + 正数 \tag{6-97}$$

由此可见 $e'^2 \geqslant e^2$。换句话说，任意 $m(x,y)$的均方误差总是至少与满足式(6-92)的 $m(x,y)$所产生的均方误差一样大。于是一个满足式(6-92)的 $m(x,y)$将使式(6-91)有最小的可能值。

式(6-92)对于 xy 平面中每个(x,y)和(α',β')可以写成下式之形式：

$$\iint m(x-\alpha,y-\beta)E\{g(\alpha,\beta)g(\alpha',\beta')\}\mathrm{d}\alpha\mathrm{d}\beta = E\{f(x,y)g(\alpha',\beta')\} \tag{6-98}$$

利用随机像场自相关和互相关的定义，对于 xy 平面中所有位置向量(x,y)和(α',β')可写成下式形式：

$$\iint m(x-\alpha,y-\beta)R_{gg}(\alpha,\beta,\alpha',\beta')\mathrm{d}\alpha\mathrm{d}\beta = R_{fg}(x,y,\alpha',\beta') \tag{6-99}$$

如果随机像场是均匀的，则其自相关函数 $R_{gg}(\alpha,\beta,\alpha',\beta')$和互相关函数 $R_{fg}(x,y,\alpha',\beta')$可表达为 $R_{gg}(\alpha-\alpha',\beta-\beta')$和 $R_{fg}(x-\alpha',y-\beta')$。所以，式(6-99)可写成下式：

$$\iint m(x-\alpha,y-\beta)R_{gg}(\alpha-\alpha',\beta-\beta')\mathrm{d}\alpha\mathrm{d}\beta = R_{fg}(x-\alpha',y-\beta') \tag{6-100}$$

为了得到一个大家习惯的标准形式，对式(6-100)中的变量作一下代换：令 $\alpha-\alpha'=t_1$，$\beta-\beta'=t_2$，$x-\alpha=\tau_1$，$y-\beta=\tau_2$，，则有 $x-\alpha=\tau_1-t_1$，$y-\beta=\tau_2-t_2$，因此，式(6-100)可写成下式形式：

$$\iint m(\tau_1-t_1,\tau_2-t_2)R_{gg}(t_1,t_2)\mathrm{d}t_1\mathrm{d}t_2 = R_{fg}(\tau_1,\tau_2)$$

再令 $t_1=x$，$t_2=y$，$\tau_1=\alpha$，$\tau_2=\beta$，则得到

$$\iint m(\alpha-x,\beta-y)R_{gg}(x,y)\mathrm{d}x\mathrm{d}y = R_{fg}(\alpha,\beta) \tag{6-101}$$

$$-\infty < \alpha < +\infty \qquad -\infty < \beta < +\infty$$

由式(6-101)可知，$m(x,y)$是恢复滤波的点扩散函数，其傅里叶变换 $M(u,v)$是传递函数。

对式(6-101)两边进行傅里叶变换，则有

$$M(u,v)S_{gg}(u,v) = S_{fg}(u,v) \tag{6-102}$$

其中 $S_{gg}(u,v)$是退化图像 $g(x,y)$的谱密度，$S_{fg}(u,v)$是退化图像与原始图像的互谱密度。由式(6-102)可见，求解最小二乘方滤波器的传递函数 $M(u,v)$需要退化图像和原始图像之间的互相关统计学知识。

如果图像 $f(x,y)$和噪声 $n(x,y)$不相关，并且 $f(x,y)$或 $n(x,y)$有零均值，则

$$E\{f(x,y)n(x,y)\} = E\{f(x,y)\}E\{n(x,y)\} = 0 \tag{6-103}$$

在这种情况下，滤波器的形式比较简单。对这种情况有

$$\begin{aligned} R_{fg}(x,y,\alpha',\beta') &= E\{f(x,y)g(\alpha',\beta')\} \\ &= \iint h(\alpha'-\alpha,\beta'-\beta)E\{f(x,y)f(\alpha,\beta)\}\mathrm{d}\alpha\mathrm{d}\beta \end{aligned} \tag{6-104}$$

考虑到随机像场的均匀性和自相关函数定义，得到

$$R_{fg}(x-\alpha',y-\beta') = \iint h(\alpha-\alpha',\beta-\beta')R_{ff}(x-\alpha,y-\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-105}$$

使用与得到式(6-101)所用的相类似的一系列变量代换，则可最后得到

$$R_{fg}(x,y) = \iint_{-\infty}^{+\infty} h(\alpha-x,\beta-y)R_{ff}(\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{6-106}$$

式(6-106)是两个确定性函数的互相关系。对两边进行傅里叶变换，得

$$S_{fg}(u,v) = H^*(u,v)S_{ff}(u,v) \tag{6-107}$$

在式(6-103)成立时，

$$S_{gg}(u,v) = S_{ff}(u,v)|H(u,v)|^2 + S_{nn}(u,v) \tag{6-108}$$

式中 $S_{nn}(u,v)$是噪声的谱密度。由此可得：

$$\begin{aligned} M(u,v) &= \frac{H^*(u,v)S_{ff}(u,v)}{S_{ff}(u,v)|H(u,v)|^2 + S_{nn}(u,v)} \\ &= \frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + [S_{nn}(u,v)/S_{ff}(u,v)]} \end{aligned} \tag{6-109}$$

由式(6-109)可见，当 $S_{nn}=0$ 时，就是理想的逆滤波器。

通常可认为噪声是白噪声，即 $S_{nn}(u,v)=$常数。若 $S_{ff}(u,v)$在 uv 平面中下降比 $S_{nn}(u,v)$快得多，这个假设就可认为是正确的。

如果有关的随机过程的统计性质不知道，也可用下式近似表示：

$$M(u,v) = \frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + \Gamma} \tag{6-110}$$

式中 Γ 是噪声对信号的功率密度比，它近似为一个适当的常数。这就是最小二乘方滤波器的传递函数。

6.4.2 用于图像复原的几种最小二乘方滤波器

除了上述的线性最小二乘方滤波器外，目前用于图像复原的还有几种变形的最小二乘方滤波器(或称为变形的维纳滤波器)。

1. 图像功率频谱滤波器

如果用 $H_R(u,v)$表示滤波器的传递函数，则图像功率频谱滤波器的传递函数有如下形式：

$$H_R(u,v) = \left[\frac{W_{FI}(u,v)}{|H_D(u,v)|^2 W_{FI}(u,v) + W_N(u,v)}\right]^{\frac{1}{2}} \tag{6-111}$$

式中 $H_D(u,v)$是图像退化的传递函数。$W_{FI}(u,v)$代表滤波器输出功率频谱，且

$$W_{FI}(u,v) = |H_D(u,v)|^2 W_{FO}(u,v) \tag{6-112}$$

式中 $W_{FO}(u,v)$代表观测的功率频谱。它与理想图像的功率频谱的关系是

$$W_{FO}(u,v) = |H_D(u,v)|^2 \cdot W_{FI}(u,v) + W_N(u,v) \tag{6-113}$$

式中 $W_N(u,v)$是噪声功率频谱。由此可见，重建图像的功率频谱和理想图像的功率频谱相同，即

$$W_{FI}(u,v) = W_{FI}(u,v) \tag{6-114}$$

2. 几何平均滤波器

几何平均滤波器的传递函数由下式表示：

$$H_R(u,v) = [H_D(u,v)]^{-S}\left[\frac{H_D^*(u,v)W_{FI}(u,v)}{|H_D(u,v)|^2 W_{FI}(u,v) + W_N(u,v)}\right]^{1-S} \tag{6-115}$$

式中 S 是一个设计参数，且 $0\leqslant S\leqslant 1$。如果 $S=\frac{1}{2}$，$H_D(u,v)=H^*(u,v)$，则几何平均滤波器与图像功率频谱滤波器相同。

3. 约束最小平方滤波器的传递函数

约束最小平方滤波器的传递函数如下：

$$H_R(u,v) = \frac{H_D^*(u,v)}{|H_D(u,v)|^2 + r|L(u,v)|^2} \tag{6-116}$$

式中 r 是一个设计常数，$L(u,v)$是一个设计频率变量。如果 $r=1$，而且使 $|L(u,v)|^2$ 等于频谱信噪功率比，那么，约束最小平方滤波器便成为标准的维纳滤波器。

6.5 约束去卷积

最小二乘方恢复滤波器或维纳滤波器是在这样的假设下推导的，即原始图像和噪声都是平稳随机场，并且它们的功率谱已知。如果没有这方面的先验知识而只知道噪声的方差的情况下，则可采用约束去卷积的方法来复原。

在一维情况下，退化模型仍然是如下形式：

$$g(x) = \int_0^T h(x-a)f(a)\mathrm{d}a + n(x) \tag{6-117}$$

式中 $g(x)$是退化信号，$f(x)$是原始信号，$h(x)$是系统冲激响应，$n(x)$是噪声项。式(6-117)的离散形式为

$$g(p) = \sum_{i=0}^{p} h(p-i)f(i) + n(p) \tag{6-118}$$

式中之 $p=0,1,2,\cdots,M+J-2$。这里假定 $f(i)$序列中有 M 个元素，$h(i)$中有 J 个元素，$g(p)$中有 $M+J-1$ 个元素。式(6-118)可写成矩阵形式：

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{n} \tag{6-119}$$

式中 $\boldsymbol{g}$、$\boldsymbol{f}$、$\boldsymbol{n}$ 分别是由 $g(p)$，$f(i)$，$n(p)$组成的向量，而 $\boldsymbol{H}$ 是一个矩阵，它的第(p,i)个元素是

$$H(p,i) = \begin{cases} h(p-i) & 0\leqslant p-i\leqslant J-1 \\ 0 & \text{其他} \end{cases} \tag{6-120}$$

此处 $p=0,1,2,\cdots,M-J-2$，$i=0,1,2,\cdots,M-1$。例如，若 $M=3$、$J=2$，则矩阵 $\boldsymbol{H}$ 取下面的形式：

$$
\boldsymbol{H}=\begin{bmatrix} h(0) & 0 & 0 \\ h(1) & h(0) & 0 \\ 0 & h(1) & h(0) \\ 0 & 0 & h(1) \end{bmatrix}
$$

在此假定

$$
e^2=\sum_{p=0}^{M+J-2} n^2(p) \tag{6-121}
$$

为一个已知的常数。要解决的恢复问题是在给定 $\boldsymbol{g}$,$\boldsymbol{H}$ 和 e^2 的情况下寻求一个 $\boldsymbol{f}$，使得

$$
[\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}]=e^2 \tag{6-122}
$$

满足式(6-122)的 $\boldsymbol{f}$ 可能很多，所以必须用其他的约束条件来选择其中最佳的 $\boldsymbol{f}$，这样一个约束条件必须具有某种先验的合理性。例如，可以用二阶导数最小作为约束条件。$f(i)$在 i 点的二阶导数可近似地用下式表示：

$$
\frac{\mathrm{d}^2 f(i)}{\mathrm{d}i^2}\approx f(i+1)-2f(i)+f(i-1)
$$

因此，选择最佳解的标准可以表达为使

$$
\sum_{i=0}^{M}[f(i+1)-2f(i)+f(i-1)]^2 \tag{6-123}
$$

最小。

如果用矩阵式表示，则式(6-123)可表达为使 $\boldsymbol{f}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{f}$ 最小，$\boldsymbol{C}$ 是下面的矩阵：

$$
\boldsymbol{C}=\begin{bmatrix}
1 & & & & & & \\
-2 & 1 & & & & \multicolumn{2}{c}{\Large 0} \\
1 & -2 & 1 & & & & \\
 & 1 & -2 & & & & \\
 & & 1 & & & & \\
 & & & \ddots & & & \\
 & & & & 1 & & \\
 & \Large 0 & & & -2 & 1 & \\
 & & & & 1 & -2 & 1 \\
 & & & & & 1 & -2 \\
 & & & & & & 1
\end{bmatrix} \tag{6-124}
$$

上述问题很自然地归入前述的约束复原方法。采用拉格朗日乘数法就可求得最小值解

令 λ 为拉格朗日乘数，则有

$$
\frac{\partial}{\partial f(i)}\{\lambda(\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f})^{\mathrm{T}}(\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f})+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{f}\}=0
$$

$$
i=0,1,2,\cdots,M-1
$$

也就是

$$
\lambda(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{f}-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g})+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{f}=0
$$

这样即可解出 $\boldsymbol{f}$：

$$
\boldsymbol{f}=\left[\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+\frac{1}{\lambda}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}\right]^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g} \tag{6-125}
$$

其中$\frac{1}{\lambda}$可用迭代法确定如下：

选一个$\frac{1}{\lambda}$值，用式(6-125)计算 $\boldsymbol{f}$，同时计算$(\boldsymbol{H}^T\boldsymbol{f}-\boldsymbol{g})^T(\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g})$，如果$\frac{1}{\lambda}$正确，则此式等于$e^2$，如果其值大于$e^2$，就减小$\frac{1}{\lambda}$；如果小于$e^2$，就增大$\frac{1}{\lambda}$，直到合适为止。约束去卷积方法应用于图像复原处理将是针对二维情况的，即

$$g(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}h(x-m,y-n)f(m,n)+n(x,y) \tag{6-126}$$

这里假定理想的原始图像矩阵 $\boldsymbol{f}$ 大小是 $M\times N$，点扩散函数矩阵 $\boldsymbol{H}$ 的大小为 $J\times K$。于是，退化图像矩阵 $\boldsymbol{g}$ 和噪声矩阵 $\boldsymbol{n}$ 的大小为$(M+J-1)\times(N+K-1)$。

二维情况下的约束方程为

$$\sum_{x=0}^{M+J-2}\sum_{y=0}^{N+K-2}n^2(x,y)=e^2 \tag{6-127}$$

相应的准则是使

$$\sum_m\sum_n[f(m-1,n)+f(m,n-1)+f(m+1,n)+f(m,n+1)-4f(m,n)]^2 \tag{6-128}$$

最小。这样可得到离散拉普拉斯算子$[L]=\begin{bmatrix}0 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & 0\end{bmatrix}$与 $\boldsymbol{f}$ 求卷积再平方。

由此可知，对于二维情况，恢复问题是在式(6-127)的约束下找到一个使式(6-128)为最小的式(6-126)的解 $\boldsymbol{f}$。

为了把问题归结到前面已讨论过的处理方法上，首先把上述公式表达为矩阵形式。其步骤如下：

第一，选择 $A\geqslant M+J-1,B\geqslant N+K-1$。形成新的延拓矩阵 $\boldsymbol{f}_e,\boldsymbol{h}_e,\boldsymbol{g}_e,\boldsymbol{n}_e$ 和 $\boldsymbol{l}_e$ 如下：

$$\boldsymbol{f}_e(x,y)=\begin{cases}\boldsymbol{f}(x,y) & \begin{cases}0\leqslant x\leqslant M-1\\ 0\leqslant y\leqslant N-1\end{cases}\\ 0 & \begin{cases}M\leqslant x\leqslant A-1\\ N\leqslant y\leqslant B-1\end{cases}\end{cases}$$

$$\boldsymbol{h}_e(x,y)=\begin{cases}\boldsymbol{h}(x,y) & \begin{cases}0\leqslant x\leqslant J-1\\ 0\leqslant y\leqslant K-1\end{cases}\\ 0 & \begin{cases}J\leqslant x\leqslant A-1\\ K\leqslant y\leqslant B-1\end{cases}\end{cases}$$

$$\boldsymbol{g}_e(x,y)=\begin{cases}\boldsymbol{g}(x,y) & \begin{cases}0\leqslant x\leqslant M+J-2\\ 0\leqslant y\leqslant N+K-2\end{cases}\\ 0 & \begin{cases}M+J-1\leqslant x\leqslant A-1\\ N+K-1\leqslant y\leqslant B-1\end{cases}\end{cases}$$

$$\boldsymbol{n}_e(x,y)=\begin{cases}\boldsymbol{n}(x,y) & \begin{cases}0\leqslant x\leqslant M+J-2\\ 0\leqslant y\leqslant N+K-2\end{cases}\\ 0 & \begin{cases}M+J-1\leqslant x\leqslant A-1\\ N+K-1\leqslant y\leqslant B-1\end{cases}\end{cases}$$

$$l_e(x,y)=\begin{cases} l(x,y) & \begin{cases}0\leqslant x\leqslant 2\\ 0\leqslant y\leqslant 2\end{cases}\\ 0 & \begin{cases}3\leqslant x\leqslant A-1\\ 3\leqslant y\leqslant B-1\end{cases}\end{cases}$$

第二，将 $\boldsymbol{f}_e,\boldsymbol{g}_e,\boldsymbol{n}_e$ 依次排列建立相应的列向量，其长度为 AB。建立的方法是使矩阵的第一行变成相应向量的第一段，第二行变成第二段，以此类推。例如，

$$\boldsymbol{f}_e=\begin{bmatrix} f_{e0}\\ \vdots\\ f_{e1}\\ \vdots\\ f_{e2}\\ \vdots\\ f_{e(A-1)}\end{bmatrix} \tag{6-129}$$

其中段 f_{ei} 是将矩阵 $\boldsymbol{f}_e$ 的第 i 行转置而形成的。用同样的方法可建立向量 $\boldsymbol{g}_e$ 和 $\boldsymbol{n}_e$。

第三，建立 $AB\times AB$ 矩阵 $\boldsymbol{H}$，$\boldsymbol{H}$ 是由 A^2 块组成，每块大小是 $B\times B$，于是有

$$\boldsymbol{H}=\begin{bmatrix} [H_0] & [H_{A-1}] & [H_{A-2}] & \cdots & [H_1]\\ [H_1] & [H_0] & [H_{A-1}] & \cdots & [H_2]\\ [H_2] & [H_1] & [H_0] & \cdots & [H_3]\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ [H_{A-1}] & [H_{A-2}] & [H_{A-3}] & \cdots & [H_0]\end{bmatrix} \tag{6-130}$$

其中每个 $[H_i]$ 是这样建立的：

$$[H_i]=\begin{bmatrix} h_e(i,0) & h_e(i,B-1) & h_e(i,B-2) & \cdots & h_e(i,1)\\ h_e(i,1) & h_e(i,0) & h_e(i,B-1) & \cdots & h_e(i,2)\\ h_e(i,2) & h_e(i,1) & h_e(i,0) & \cdots & h_e(i,3)\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ h_e(i,B-1) & h_e(i,B-2) & h_e(i,B-3) & \cdots & h_e(i,0)\end{bmatrix} \tag{6-131}$$

同样方法从矩阵 $\boldsymbol{l}_e$ 可建立 $AB\times AB$ 矩阵 $\boldsymbol{L}$。

按照上面定义的向量 $\boldsymbol{f}_e,\boldsymbol{g}_e,\boldsymbol{n}_e$ 和矩阵 $\boldsymbol{H}$ 及 $\boldsymbol{L}$，可将式(6-126)、(6-127)和(6-128)表示为如下矩阵式形式：

$$\boldsymbol{g}_e=\boldsymbol{H}\,\boldsymbol{f}_e+\boldsymbol{n}_e \tag{6-132}$$

$$(\boldsymbol{g}_e-\boldsymbol{H}\,\boldsymbol{f}_e)^{\mathrm{T}}(\boldsymbol{g}_e-\boldsymbol{H}\,\boldsymbol{f}_e)=e^2 \tag{6-133}$$

并且使

$$\boldsymbol{f}_e^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\,\boldsymbol{f}_e \tag{6-134}$$

为最小。

这样的恢复问题就是要找一个使式(6-134)最小的式(6-132)的解 $\boldsymbol{f}_e$，并且同时满足式(6-133)给出的约束条件。

由前面的讨论，这一解可直接由下式得出：

$$\boldsymbol{f}_e=\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+\frac{1}{\lambda}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{g}_e \tag{6-135}$$

这里 $A\geqslant M+2J-2, B\geqslant N+2K-2$。

直接求解式(6-135)比较困难。可以用傅里叶变换的方法在变换域中计算式(6-135)。由于

矩阵 $\boldsymbol{H}$ 和 $\boldsymbol{L}$ 是分块循环矩阵，所以二维傅里叶变换可把分块循环矩阵变成对角形矩阵。

设 $\boldsymbol{W}$ 为一个 $AB \times AB$ 矩阵，它由 A^2 块组成，每块大小为 $B \times B$。$\boldsymbol{W}$ 的第(m,n)块记为$[W]_{mn}$并表示如下：

$$
\begin{gathered}
[W]_{mn} = \exp\left(\mathrm{j}\,\frac{2\pi}{A}mn\right)[W] \\
m,n = 0,1,2,\cdots,A-1
\end{gathered}
\tag{6-136}
$$

其中$[W]$是 $B \times B$ 矩阵，其第(i,k)元由下式表示：

$$
\begin{gathered}
W(i,k) = \exp\left(\mathrm{j}\,\frac{2\pi}{B}ik\right) \\
i,k = 0,1,2,\cdots,B-1
\end{gathered}
\tag{6-137}
$$

矩阵 $\boldsymbol{W}$ 的逆矩阵 $\boldsymbol{W}^{-1}$也由 A^2 块组成，每块大小也是 $B \times B$。如果用$[W]_{mn}^{-1}$表示矩阵$[W]^{-1}$的第(m,n)块，则有

$$
\begin{gathered}
[W]_{mn}^{-1} = \frac{1}{A}\exp\left(-\mathrm{j}\,\frac{2\pi}{A}mn\right)[W]^{-1} \\
m,n = 0,1,2,\cdots,A-1
\end{gathered}
\tag{6-138}
$$

式中$[W]^{-1}$是 $B \times B$ 矩阵，其第(i,k)个元素由下式表示：

$$
\begin{gathered}
W^{-1}(i,k) = \frac{1}{B}\exp\left(-\mathrm{j}\,\frac{2\pi}{B}ik\right) \\
i,k = 0,1,2,\cdots,B-1
\end{gathered}
\tag{6-139}
$$

令$[\boldsymbol{F}_e]$，$[\boldsymbol{H}_e]$，$[\boldsymbol{G}_e]$，$[\boldsymbol{L}_e]$分别是 $\boldsymbol{f}_e,\boldsymbol{h}_e,\boldsymbol{g}_e,\boldsymbol{l}_e$ 的二维离散傅里叶变换。其向量仍然由形成$\boldsymbol{f}_e$那样的方法形成$\boldsymbol{f}_e$，则有

$$
\begin{aligned}
[\boldsymbol{F}_e] &= \boldsymbol{W}^{-1}\boldsymbol{f}_e \\
[\boldsymbol{G}_e] &= \boldsymbol{W}^{-1}\boldsymbol{g}_e \\
\boldsymbol{H} &= \boldsymbol{W}\boldsymbol{D}_h\boldsymbol{W}^{-1} \\
\boldsymbol{H}^{T} &= \boldsymbol{W}\boldsymbol{D}_h^{*}\boldsymbol{W}^{-1} \\
\boldsymbol{L} &= \boldsymbol{W}\boldsymbol{D}_l\boldsymbol{W}^{-1} \\
\boldsymbol{L}^{T} &= \boldsymbol{W}\boldsymbol{D}_l^{*}\boldsymbol{W}^{-1}
\end{aligned}
\tag{6-140}
$$

式中 $\boldsymbol{D}_h$ 和 $\boldsymbol{D}_l$ 是 $AB \times AB$ 对角形矩阵。对角形矩阵的元由下式表示：

$$
\begin{gathered}
D_h(k,k) = ABH_e\left(\left[\frac{k}{A}\right], k \bmod B\right) \\
D_l(k,k) = ABL_e\left(\left[\frac{k}{A}\right], k \bmod B\right) \\
k = 0,1,2,\cdots,AB-1
\end{gathered}
\tag{6-141}
$$

式中$\left[\frac{k}{A}\right]$表示小于$\frac{k}{A}$的最大整数，$k \bmod B$ 是 k 除以 B 所得的余数。矩阵 $\boldsymbol{H}_e$ 和 $\boldsymbol{L}_e$ 前面已有定义，$\boldsymbol{H}_e(u,v)$和 $\boldsymbol{L}_e(u,v)$分别是$[\boldsymbol{H}_e]$和$[\boldsymbol{L}_e]$的二维离散傅里叶变换。

把式(6-140)中的后四个关系式代入式(6-135)，利用式(6-140)前两式及向量二维离散傅里叶变换的关系可得到：

$$
\begin{gathered}
F_e(u,v) = \frac{1}{AB}\,\frac{H_e^{*}(u,v)G_e(u,v)}{|H_e(u,v)|^2 + \frac{1}{\lambda}|L_e(u,v)|^2} \\
u = 0,1,2,\cdots,A-1 \qquad v = 0,1,2,\cdots,B-1
\end{gathered}
\tag{6-142}
$$

在复原过程中所用的滤波器传递函数为

$$M(u,v)=\frac{1}{H_e(u,v)}\frac{|H_e(u,v)|^2}{AB\left\{|H_e(u,v)|^2+\frac{1}{\lambda}|L_r(u,v)|^2\right\}} \tag{6-143}$$

$$u=0,1,2,\cdots,A-1 \qquad v=0,1,2,\cdots,B-1$$

这个公式有点类似于维纳滤波器，但是两者之间有重大区别。维纳滤波器是对一族图像在平均意义上的最好复原，而这一公式只对一幅图像给出最佳复原。另外，推导维纳滤波器的基本假定的随机像场是均匀的并且谱密度为已知，而此处滤波器没有作这样的假设而只是确定了一个最佳准则。

6.6 中值滤波

对受到噪声污染的退化图像的复原可以采用线性滤波方法来处理，有许多情况下是很有效的。但是多数线性滤波具有低通特性，在去除噪声的同时也使图像的边缘变得模糊了。中值滤波方法在某些条件下可以作到既去除噪声又保护图像边缘的较满意的复原。中值滤波是一种去除噪声的非线性处理方法，它是由图基(Turky)在1971年提出的。开始，中值滤波用于时间序列分析，后来被用于图像处理，在去噪复原中得到了较好的效果。

6.6.1 中值滤波的基本原理

中值滤波的基本原理是把数字图像或数字序列中一点的值用该点的一个邻域中各点值的中值代替。中值的定义如下：

一组数 $x_1,x_2,x_3,\cdots,x_n$，把 n 个数按值的大小顺序排列于下：

$$x_{i1}\leqslant x_{i2}\leqslant x_{i3}\leqslant\cdots\leqslant x_{in}$$

$$y=\text{Med}\{x_1,x_2,x_3,\cdots,x_n\}=\begin{cases}x_{i(\frac{n+1}{2})} & n\text{ 为奇数}\\ \frac{1}{2}\left[x_{i(\frac{n}{2})}+x_{i(\frac{n}{2}+1)}\right] & n\text{ 为偶数}\end{cases} \tag{6-144}$$

y 称为序列 $x_1,x_2,x_3,\cdots,x_n$ 的中值。例如有一序列为(80, 90, 200, 110, 120)，这个序列的中值为110。

把一个点的特定长度或形状的邻域称作窗口。在一维情形下，中值滤波器是一个含有奇数个像素的滑动窗口。窗口正中间那个像素的值用窗口内各像素值的中值代替。

设输入序列为$\{x_i,i\in I\}$，I 为自然数集合或子集，窗口长度为 n。则滤波器输出为

$$y_i=\text{Med}\{x_i\}=\text{Med}\{x_{i-u}\cdots x_i\cdots x_{i-u}\} \tag{6-145}$$

其中 $i\in I$，$u=\dfrac{(n-1)}{2}$。

例如，有一输入序列如下：

$$\{x_i\}=\{0\ 0\ 0\ 8\ 0\ 0\ 2\ 3\ 2\ 0\ 2\ 3\ 2\ 0\ 3\ 5\ 3\ 0\ 3\ 5\ 3\ 0\ 0\ 2\ 3\ 4\ 5\ 5\ 5\ 5\ 5\ 0\ 0\ 0\}$$

在此序列中前面的8是脉冲噪声，中间一段是一种寄生振荡，后面是希望保留的斜坡和跳变。在此采用长度为3的窗口，得到的结果为

$$\{y_i\}=\{0\ 0\ 0\ 0\ 0\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 0\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 0\ 0\ 2\ 3\ 4\ 5\ 5\ 5\ 5\ 5\ 0\ 0\ 0\ 0\}$$

显然，经中值滤波后，脉冲噪声8被滤除了，振荡被平滑掉了，斜坡和阶跃部分被保存了下来。

中值滤波的运算方法可以在有限程度上作些分析。例如，常数 K 与序列 $f(i)$ 相乘的中值

有如下关系存在：

$$\mathrm{Med}\{Kf(i)\} = K\mathrm{Med}\{f(i)\} \tag{6-146}$$

而常数 K 与序列 $f(i)$相加的中值有如下关系：

$$\mathrm{Med}\{K + f(i)\} = K + \mathrm{Med}\{f(i)\} \tag{6-147}$$

对几种基本信号进行中值滤波的例子如图 6-4 所示。图中(a)是阶跃信号，经中值滤波后仍然保持了阶跃部分；图(b)原始信号是斜坡，滤波后也保持了其形状；图(c)的原始信号是单脉冲信号，经滤波后消去了这个脉冲；图(d)中的原始信号是双脉冲，经中值滤波后也被消去了；图(e)的原始信号是三脉冲，滤波后对其没有影响；图(f)的原始信号是三角形，滤波后虽然有少许变形，但也还基本保持了原来的形状。

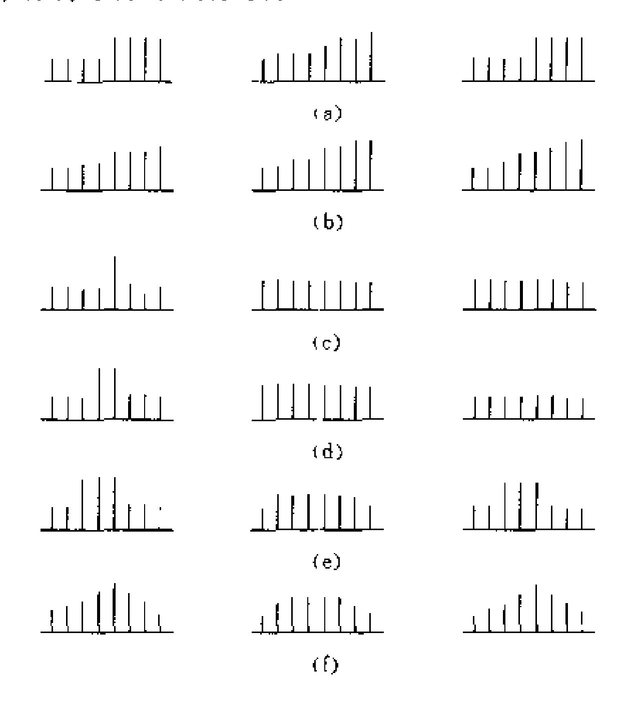


图 6-4 对几种基本信号中值滤波的结果的例子

中值滤波的概念很容易推广到二维，此时可以利用某种形式的二维窗口。设$\{x_{ij},(i,j)\in I^2\}$表示数字图像各点的灰度值，滤波窗口为 A 的二维中值滤波可定义为

$$y_{ij} = \underset{A}{\mathrm{Med}}\{x_{ij}\} = \mathrm{Med}\{x_{(i+r),(j-s)},(r,s) \in A,(i,j) \in I^2\} \tag{6-148}$$

二维中值滤波的窗口可以取方形，也可以取近似圆形或十字形。

图 6-5 是二维中值滤波的实例。图中(a)是原始图像，图(b)是混有高斯白噪声的图像，(c)是 3×3 窗口中值滤波结果图像，(d)是 5×5 窗口中值滤波结果图像，(e)是 3×3 窗口均值滤波结果图像，(f)是 5×5 窗口均值滤波结果图像，(g)是加有椒盐噪声的图像，(h)是 3×3 窗口中值滤波结果图像，(i)是 5×5 窗口中值滤波结果图像，(j)是 3×3 窗口均值滤波结果图像，(k)是采用 5×5 窗口均值结果图像。



(a) 原像



(b) 高斯白噪声图像



(c) 3×3 中值滤波图像



(d) 5×5 中值滤波图像



(e) 3×3 均值滤波图像



(f) 5×5 均值滤波图像



(g) 加有椒盐噪声图像



(h) 3×3 中值滤波图像



(i) 5×5 中值滤波图像



(j) 3×3 均值滤波图像



(k) 5×5 均值滤波图像

图 6-5 二维中值滤波及均值滤波实例

6.6.2 加权的中值滤波

以上讨论中的中值滤波，窗口内各点对输出的作用是相同的。如果希望强调中间点或距中间点最近的几个点的作用，可以采用加权中值滤波法。加权中值滤波的基本原理是改变窗口中变量的个数，可以使一个以上的变量等于同一点的值，然后对扩张后的数字集求中值。以窗口为 3 的一维加权中值滤波为例，表示如下：

$$\begin{aligned} y_i &= \text{Weighted_Med}\{x_{i-1}, x_i, x_{i+1}\} \\ &= \text{Med}\{x_{i-1}, x_{i-1}, x_i, x_i, x_i, x_{i+1}, x_{i+1}\} \end{aligned} \tag{6-149}$$

由公式(6-149)可见，在窗口内，中间点取奇数，两边点取对称数，也就是位于窗口中间的像素重复两次，位于窗口边缘的两个像素重复一次，形成新的序列，然后对新的序列再施以常规中值滤波处理。

二维加权中值滤波与一维情况类似。如果适当地选取窗口内各点的权重，加权中值滤波比简单中值滤波能更好地从受噪声污染的图像中恢复出阶跃边缘以及其他细节。二维加权中值滤波以 3×3 窗口为例，表示如下：

原始窗口为

$$\begin{matrix} x_{i-1,j-1} & x_{i-1,j} & x_{i-1,j+1} \\ x_{i,j-1} & x_{i,j} & x_{i,j+1} \\ x_{i+1,j-1} & x_{i+1,j} & x_{i+1,j+1} \end{matrix}$$

加权后的中值滤波如下式所示：

$$
\begin{aligned}
y_{ij} &= \text{Weighted_Med}\{x_{i-1,j-1}, x_{i-1,j}, x_{i-1,j+1}, x_{i,j-1}, x_{i,j}, x_{i,j+1}, x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j+1}\} \\
&= \text{Med}\{x_{i-1,j-1}, x_{i-1,j}, x_{i-1,j}, x_{i-1,j+1}, x_{i,j-1}, x_{i,j-1}, x_{i,j}, \\
&\quad x_{i,j}, x_{i,j}, x_{i,j-1}, x_{i,j-1}, x_{i+1,j-1}, x_{i-1,j}, x_{i+1,j}, x_{i-1,j+1}\}
\end{aligned} \tag{6-150}
$$

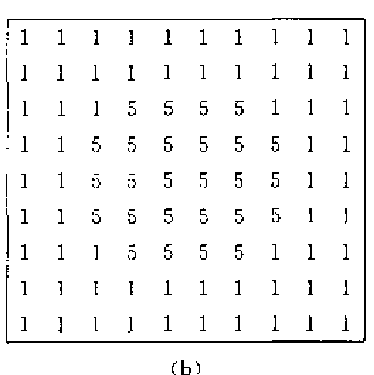
即中间的点取三个值(重复两次),上、下、左、右的点各取两个(重复一次),对角线上的点取一个(不重复)。

加权中值滤波与普通中值滤波有时会有不同的效果。例如,对于普通中值滤波有 $y=\text{Med}\{1\ 1\ 1\ 1\ 5\ 5\ 1\ 5\ 5\}=1$;而加权后的中值滤波为 $y=\text{Med}\{1\ 1\ 1\ 1\ 1\ 5\ 5\ 5\ 5\ 1\ 5\ 5\ 5\}=5$。加权中值滤波保持了方块角上的一点的值。

中值滤波可有效地去除脉冲型噪声,而且对图像的边缘有较好的保护。但是它也有其固有的缺陷,如果使用不当,会损失许多图像细节。例如,采用 3×3 窗口对图 6-6(a)所示的原始图像滤波,滤波结果如图(b)所示,其结果不但削去了方块的 4 个角,而且把中间的小方块也滤掉了。因此,中值滤波在选择窗口时要考虑其形状及等效带宽,以避免滤波处理造成的信息损失。图 6-7 是中值滤波的另一实例。图(a)是一条细线条图像,经 3×3 窗口滤波后,图像中的细线条完全滤掉了,如图(b)所示。以上两例可以直观地看到,中值滤波对图像中的细节处很不理想,所以,中值滤波对所谓的椒盐噪声(pepper noise)的滤除非常有效,但是它对点、线等细节较多的图像却不太适用。

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	8	8	5	5	1	1
1	1	5	5	8	8	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

(a)



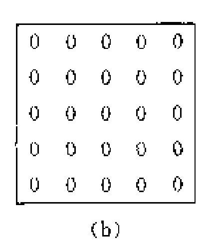
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	5	5	5	5	1	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	1	5	5	5	5	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

(b)

图 6-6　中值滤波实例一

0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

(a)



0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(b)

图 6-7　中值滤波实例二

本章给出一个中值和均值滤波的计算机程序(见附录 8),以供读者参考。

在图 6-4 中,为了比较中值滤波的效果,也给出了均值滤波的处理结果。均值滤波的滤波过程也是使一个窗口在图像(或序列)上滑动,窗中心位置的值用窗内各点值的平均值来代替。以二维均值滤波为例,它的定义如下:

设 $\{x_{ij}\}$ 表示数字图像各像素的灰度值,A 为一个 3×3 的窗口,则二维均值滤波的定义为

$$
\begin{aligned}
y_{ij} = \text{mean}\{x_{ij}\} = \frac{1}{9}\{&x_{i-1,j-1} + x_{i-1,j} + x_{i-1,j-1} + x_{i,j-1} + x_{i,j} \\
&+ x_{i,j+1} + x_{i+1,j-1} + x_{i+1,j} + x_{i-1,j+1}\}
\end{aligned} \tag{6-151}
$$

一般均值滤波的边缘保护特性不如中值滤波。

6.7 几种其他空间复原技术

前边讨论了几种基本的代数图像复原技术。除此之外，尚有一些其他的空间图像复原方法，本节将对这些方法作一些简要的讨论。

6.7.1 几何畸变校正

在图像的获取或显示过程中往往会产生几何失真。例如，成像系统有一定的几何非线性。这主要是由于视像管摄像机及阴极射线管显示器的扫描偏转系统有一定的非线性，因此会造成如图 6-8 所示的枕形失真或桶形失真。图(a)为原始图像，图(b)和图(c)为失真图像。除此之外还有由于斜视角度获得的图像的透视失真。另外，由卫星摄取的地球表面的图像往往覆盖较大的面积，由于地球表面呈球形，这样摄取的平面图像也将会有较大的几何失真。对于这些图像必须加以校正，以免影响分析精度。

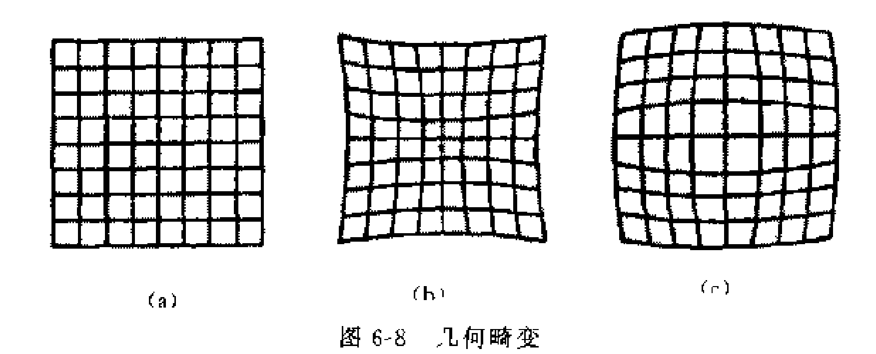


图 6-8 几何畸变

由成像系统引起的几何畸变的校正有两种方法。一种是预畸变法，这种方法是采用与畸变相反的非线性扫描偏转法，用来抵消预计的图像畸变；另一种是所谓的后验校正方法。这种方法是用多项式曲线在水平和垂直方向去拟合每一畸变的网线，然后求得反变化的校正函数。用这个校正函数即可校正畸变的图像。图像的空间几何畸变及其校正过程如图 6-9 所示。

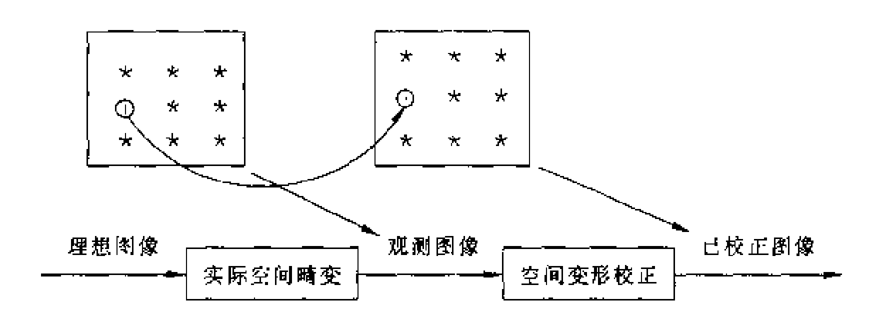


图 6-9 空间几何畸变及校正的概念

任意几何失真都可由非失真坐标系(x,y)变换到失真坐标系(x',y')的方程来定义。方程的一般形式为

$$\begin{cases} x' = h_1(x,y) \\ y' = h_2(x,y) \end{cases} \tag{6-152}$$

在透视畸变的情况下，变换是线性的，即

$$\begin{cases} x' = ax + by + c \\ y' = dx + ey + f \end{cases} \tag{6-153}$$

设 $f(x,y)$ 是无失真的原始图像，$g(x',y')$ 是 $f(x,y)$ 畸变的结果，这一失真的过程是已知的并且用函数 h_1 和 h_2 定义。于是有：

$$g(x',y') = f(x,y) \tag{6-154}$$

这说明在图像中本来应该出现在像素 (x,y) 上的灰度值由于失真实际上却出现在 (x',y') 上了。这种失真的复原问题实际上是映射变换问题。在给定了 $g(x',y')$，$h_1(x,y)$，$h_2(x,y)$ 的情况下，其复原处理可如下进行：

1) 对于 $f(x,y)$ 中的每一点 (x_0,y_0)，找出在 $g(x',y')$ 中相应的位置 $(\alpha,\beta)=[h_1(x_0,y_0),h_2(x_0,y_0)]$。由于 α 和 β 不一定是整数，所以通常 (α,β) 不会与 $g(x',y')$ 中的任何点重合。

2) 找出 $g(x',y')$ 中与 (α,β) 最靠近的点 (x'_1,y'_1)，并且令 $f(x_0,y_0)=g(x'_1,y'_1)$，也就是把 $g(x'_1,y'_1)$ 点的灰度值赋于 $f(x_0,y_0)$。如此逐点作下去，直到整个图像，则几何畸变得到校正。

3) 如果不采用 2) 中的灰度值的代换方法也可以采用内插法。这种方法是假定 (α,β) 点找到后，在 $g(x',y')$ 中找出包围着 (α,β) 的 4 个邻近的数字点，(x'_1, y'_1)，(x'_{1+1},y'_1)，(x'_1,y'_{1+1})，(x'_{1+1},y'_{1+1}) 并且有：

$$\begin{cases} x'_1 \leqslant \alpha < x'_{1+1} \\ y'_1 \leqslant \beta < y'_{1+1} \end{cases} \tag{6-155}$$

$f(x,y)$ 中点 (x_0,y_0) 的灰度值由 $g(x',y')$ 中 4 个点的灰度值间的某种内插法来确定。

在以上方法的几何校正处理中，如果 (α,β) 处在图像 $g(x',y')$ 之外，则不能确定其灰度值，而且校正后的图像多半不能保持其原来的矩形形状。

以上讨论的是 g,h_1,h_2 都知道的情况下几何畸变的校正方法。如果只知道 g，而 h_1 和 h_2 都不知道，但是若有类似规则的网格之类的图案可供参考利用，那么就有可能通过测量 g 中的网格点的位置来决定失真变换的近似值。

例如，如果给出了三个邻近网格点构成的小三角形，其在规则网格中的理想坐标为 (r_1,s_1)，(r_2,s_2)，(r_3,s_3)，并设这些点在 g 中的位置分别为 (u_1,v_1)，(u_2,v_2)，(u_3,v_3)。由线性变换关系：

$$\begin{cases} x' = ax + by + c \\ y' = dx + ey + f \end{cases}$$

可认为它把三个点映射到它们失真后的位置，由此，可构成如下 6 个方程：

$$\begin{cases} u_1 = ar_1 + bs_1 + c \\ v_1 = dr_1 + es_1 + f \\ u_2 = ar_2 + bs_2 + c \\ v_2 = dr_2 + es_2 + f \\ u_3 = ar_3 + bs_3 + c \\ v_3 = dr_3 + es_3 + f \end{cases} \tag{6-156}$$

解这 6 个方程可求得 a,b,c,d,e,f。这种变换可用来校正 g 中被这三点联线包围的三角形部分的失真。由此对每三个一组的网格点重复进行，即可实现全部图像的几何校正。

6.7.2 盲目图像复原

多数的图像复原技术都是以图像退化的某种先验知识为基础，也就是假定系统的脉冲响

应是已知的。但是,在许多情况下难以确定退化的点扩散函数。在这种情况下,必须从观察图像中以某种方式抽出退化信息,从而找出图像复原方法。这种方法就是所谓的盲目图像复原。对具有加性噪声的模糊图像作盲目图像复原的方法有两种,就是直接测量法和间接估计法。

直接测量法盲目图像复原通常要测量图像的模糊脉冲响应和噪声功率谱或协方差函数。在所观察的景物中,往往点光源能直接指示出冲激响应。另外,图像边缘是否陡峭也能用来推测模糊冲激响应。在背景亮度相对恒定的区域内测量图像的协方差可以估计出观测图像的噪声协方差函数。

间接估计法盲目图像复原类似于多图像平均法处理。例如,在电视系统中,观测到的第 i 帧图像为

$$g_i(x,y) = f_i(x,y) + n_i(x,y) \tag{6-157}$$

式中 $f_i(x,y)$ 是原始图像, $g_i(x,y)$ 是含有噪声的图像, $n_i(x,y)$ 是加性噪声。如果原始图像在 M 帧观测图像内保持恒定,对 M 帧观测图像求和,得到下式之关系:

$$f_i(x,y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x,y) - \frac{1}{M}\sum_{i=1}^{M} n_i(x,y) \tag{6-158}$$

当 M 很大时,式(6-158)右边的噪声项的值趋向于它的数学期望值 $E\{n(x,y)\}$。一般情况下白色高斯噪声在所有 (x,y) 上的数学期望等于零,因此,合理的估计量是:

$$\hat{f}_i(x,y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x,y) \tag{6-159}$$

盲目图像复原的间接估计法也可以利用时间上平均的概念去掉图像中的模糊。如果有一成像系统,其中相继帧含有相对平稳的目标退化,这种退化是由于每帧有不同的线性位移不变冲激响应 $h_i(x,y)$ 引起的。例如,大气湍流对远距离物体摄影就会产生这种图像退化。只要物体在帧间没有很大移动并每帧取短时间曝光,那么第 i 帧的退化图像可表示为

$$g_i(x,y) = f_i(x,y) * h_i(x,y) \tag{6-160}$$

式中 $f_i(x,y)$ 是原始图像, $g_i(x,y)$ 是退化图像, $h_i(x,y)$ 是点扩散函数, $*$ 代表卷积。式中 $i=1,2,\cdots,M$。退化图像的傅里叶变换为

$$G_i(u,v) = F_i(u,v)H_i(u,v) \tag{6-161}$$

利用同态处理方法把原始图像的频谱和退化传递函数分开,则可得到:

$$\ln[G_i(u,v)] = \ln[F_i(u,v)] + \ln[H_i(u,v)] \tag{6-162}$$

如果帧间退化冲激响应是不相关的,则可得到下面的和式:

$$\sum_{i=1}^{M}\ln[G_i(u,v)] = M\ln[F_i(u,v)] + \sum_{i=1}^{M}\ln[H_i(u,v)] \tag{6-163}$$

当 M 很大时,传递函数的对数和式接近于一恒定值,即

$$K_{\rm H}(u,v) = \lim_{M\to\infty}\sum_{i=1}^{M}\ln[H_i(u,v)] \tag{6-164}$$

因此,图像的估计量为

$$\hat{F}_i(u,v) = \exp\left\{\frac{K_{\rm H}(u,v)}{M}\right\}\prod_{i=1}^{M}[G_i(u,v)]^{\frac{1}{M}} \tag{6-165}$$

对式(6-165)取傅里叶反变换就可得到空域估计是 $\hat{f}(x,y)$。

在上面分析中,并没考虑加性噪声分量。如果考虑加性噪声分量,则无法进行式(6-162)的分离处理,后边的推导也就不成立了。对于这样的问题,可以对观测到的每帧图像先进行滤波处理、去掉噪声,然后在图像没有噪声的假设下再进行上述处理。

6.7.3 递归图像复原技术

递归图像复原技术是将递归估计技术用于图像复原处理的一种方法。递归估计技术是1960年左右发展起来的。最初多用于时域信号递归滤波处理，这就是卡尔曼滤波技术。近年来已把这种技术直接用于图像复原处理中。

1. 广义马尔可夫序列与卡尔曼滤波

设有两个随机变量序列 $s_1, s_2, s_3, \cdots, s_n$ 和 $x_1, x_2, x_3, \cdots, x_n$。第一个序列称为信号序列，第二个称为数据样本序列。把 $\boldsymbol{x}_n$ 写做：

$$\boldsymbol{x}_n = \boldsymbol{s}_n + \boldsymbol{v}_n \tag{6-166}$$

式中 $\boldsymbol{v}_n$ 是噪声。可以用数据 $x_1, x_2, x_3, \cdots, x_n$，的线性组合来估计 $\boldsymbol{s}_n$，即

$$\hat{s}_n = \alpha_1^n x_1 + \alpha_2^n x_2 + \cdots + \alpha_n^n x_n \tag{6-167}$$

由正交性原理，当差值 $\boldsymbol{s}_n - \hat{\boldsymbol{s}}_n$ 对各个数据样本 x_i 为正交时，所解出的各个系数 α_i^n 可使均方误差最小，即

$$E\left\{(\boldsymbol{s}_n - \hat{\boldsymbol{s}}_n)\boldsymbol{x}_i\right\} = 0 \qquad i = 1, 2, \cdots, m \tag{6-168}$$

使

$$E\{[\boldsymbol{s}_n - \hat{\boldsymbol{s}}_n]^2\} = \min \tag{6-169}$$

式中 E 代表数学期望。

如果矩量 $E\{\boldsymbol{s}_n\boldsymbol{x}_i\}$ 和 $E\{\boldsymbol{x}_k\boldsymbol{x}_i\}$ 已知，则由上面的方程式组可解出未知系数 α_i^n。这时，所得到的均方估计误差为

$$e_n = E\{(\boldsymbol{s}_n - \hat{\boldsymbol{s}}_n)^2\} = E\{(\boldsymbol{s}_n - \hat{\boldsymbol{s}}_n)\boldsymbol{s}_n\} \tag{6-170}$$

由此可见，为了估计 $\boldsymbol{s}_{n+1}$，就需要确定一组 $n+1$ 个新的系数 α_i^{n+1}。

如果一个随机变量序列 $\boldsymbol{s}_n$，用它前边的所有的随机变量 $\boldsymbol{s}_{n-1}, \boldsymbol{s}_{n-2}, \cdots$ 等对它所作的线性均方估计与仅用 $\boldsymbol{s}_{n-1}$ 时的估计相同，则这个随机变量序列 $\boldsymbol{s}_n$ 就叫做广义马尔可夫序列。

从矢量投影的观点来看，相当于在 $\boldsymbol{s}_{n-1}, \boldsymbol{s}_{n-2}, \cdots$ 空间内，$\boldsymbol{s}_n$ 的投影只与 $\boldsymbol{s}_{n-1}$ 符合。由相关函数的概念：

$$\boldsymbol{R}_{ik} = E\{\boldsymbol{s}_i\boldsymbol{s}_k\} \tag{6-171}$$

及

$$\hat{\boldsymbol{s}}_n = A_n\boldsymbol{s}_{n-1} \tag{6-172}$$

则有：

$$E\{(\boldsymbol{s}_n - A_n\boldsymbol{s}_{n-1})\boldsymbol{s}_{n-1}\} = 0$$

$$A_n = \frac{R_{n-1,n}}{R_{n-1,n-1}} \tag{6-173}$$

设信号 $\boldsymbol{s}_n$ 是广义马尔可夫序列，噪声 $\boldsymbol{v}_n$ 由正交的随机变量组成，并对 $\boldsymbol{s}_i$ 也是正交的，则有下列各式成立：

$$\boldsymbol{x}_i = \boldsymbol{s}_i + \boldsymbol{v}_i \tag{6-174}$$

$$E\{\boldsymbol{v}_i^2\} = \sigma_i^2 \tag{6-175}$$

$$E\{\boldsymbol{s}_i, \boldsymbol{v}_n\} = 0 \qquad i, n = 1, 2, \cdots \tag{6-176}$$

因此，可写出

$$E\{\boldsymbol{s}_i, \boldsymbol{x}_n\} = E\{\boldsymbol{s}_i, \boldsymbol{s}_n\} = R_{in} \tag{6-177}$$

$$E\{x_i, x_n\} = \begin{cases} R_n + \sigma_i^2 & i = n \\ R_{in} & i \neq n \end{cases} \tag{6-178}$$

广义马尔可夫序列递推滤波的原理如下：

用 $x_n, x_{n-1}, \cdots, x_1$ 对 s_n 的线性估计 $\hat{s}_n$ 用下式表示：

$$\hat{s}_n = \alpha_n \hat{s}_{n-1} + \beta_n x_n \tag{6-179}$$

式中 $\hat{s}_{n-1}$是对 s_{n-1}的估计，α_n 和 β_n 是二个待定的常数。

问题的要点是，因为 $\hat{s}_n$是用 $x_n, x_{n-1}, \cdots$对 s_n的估计，所以必有以下二式成立：

$$E\{(s_n - \hat{s}_n)x_n\} = 0 \tag{6-180}$$

$$E\{(s_n - \hat{s}_n)x_i\} = 0 \qquad i = 1, 2, \cdots, n-1 \tag{6-181}$$

显然式(6-179)能成立的充分条件是能够找到满足式(6-180)和式(6-181)的 α_n 和 β_n。

由式(6-173)的 A_n 可知，$s_n - A_n s_{n-1}$与 $s_{n-1}, s_{n-2}, \cdots, s_1$ 是正交的，并且 s_i 与 v_n 也正交，所以

$$E\{(s_n - A_n \hat{s}_{n-1})v_i\} = 0 \qquad i = 1, 2, \cdots, n-1 \tag{6-182}$$

将 $s_n - \hat{s}_n$写为如下形式：

$$\begin{aligned} s_n - \hat{s}_n &= s_n - \alpha_n \hat{s}_{n-1} - \beta_n x_n \\ &= s_n - \alpha_n s_{n-1} - \beta_n s_n + \alpha_n (s_{n-1} - \hat{s}_{n-1}) - \beta_n v_n \end{aligned} \tag{6-183}$$

因为 $\hat{s}_{n-1}$是 s_{n-1}的估计，所以$(s_{n-1} - \hat{s}_{n-1})$对 $x_{n-1}, x_{n-2}, \cdots, x_1$ 正交。并且 v_n 与 $x_{n-1}, x_{n-2}, \cdots, x_1$ 也正交。为了使式(6-181)成立，则必须有下式成立：

$$E\{[s_n(1 - \beta_n) - \alpha_n s_{n-1}]x_i\} = 0 \qquad i = 1, 2, \cdots, n-1 \tag{6-184}$$

由式(6-182)可知，只要使

$$\frac{\alpha_n}{1 - \beta_n} = A_n = \frac{R_{n-1,n}}{R_{n-1,n-1}} \tag{6-185}$$

则式(6-182)可成立。由此，式(6-181)得以成立。问题是如何确定 α_n 和 β_n。因为

$$\begin{aligned} E\{(s_n - \hat{s}_{n-1})x_n\} &= E\{(s_n - \alpha_n \hat{s}_{n-1} - \beta_n x_n)x_n\} \\ &= E\{x_n s_n - \alpha_n x_n \hat{s}_{n-1} - \beta_n x_n x_n\} = 0 \end{aligned}$$

由式(6-177)和式(6-178)可得到：

$$R_{nn} = \alpha_n E\{x_n \hat{s}_{n-1}\} + \beta_n (R_{nn} + \sigma_n^2) \tag{6-186}$$

$$E\{x_n \hat{s}_{n-1}\} = E\{(s_n + v_n)\hat{s}_{n-1}\} = E\{s_n \hat{s}_{n-1}\} \tag{6-187}$$

其均方估计误差可求得如下：

$$\begin{aligned} e_n &= E\{(s_n - \alpha_n \hat{s}_{n-1} - \beta_n x_n)s_n\} \\ &= R_{nn} - \alpha_n E\{s_n \hat{s}_{n-1}\} - \beta_n R_{nn} \\ &= \beta_n \sigma_n^2 \end{aligned}$$

又由

$$E\{(s_n - A_n s_{n-1})\hat{s}_{n-1}\} = E\{(s_n \hat{s}_{n-1}\} - A_n E\{s_{n-1}\hat{s}_{n-1}\} = 0$$

$$e_{n-1} = E\{(s_{n-1} - \hat{s}_{n-1})s_{n-1}\} = R_{n-1,n-1} - E\{s_{n-1}\hat{s}_{n-1}\}$$

可写出

$$E\{s_n \hat{s}_{n-1}\} = A_n (R_{n-1,n-1} - e_{n-1}) \tag{6-188}$$

于是得到

$$R_{nn} = \alpha_n A_n (R_{n-1n-1} - e_{n-1}) + \beta_n (R_{nn} + \sigma_n^2)$$

$$\beta_n = \frac{e_n}{\sigma_n^2}, \qquad \alpha_n = A_n \left(1 - \frac{e_n}{\sigma_n^2}\right) \tag{6-189}$$

最后解得

$$e_n = \frac{R_{nn} - A_n^2 R_{n-1,n-1} + A_n^2 e_{n-1}}{R_{nn} - A_n^2 R_{n-1,n-1} + A_n^2 e_{n-1} + \sigma_n^2}\sigma_n^2 \tag{6-190}$$

由上面分析可见，卡尔曼滤波的计算步骤可归结于下：在 R_{ik} 与 σ_i^2 已知时，在第 $n-1$ 步中已计算出 $\hat{s}_{n-1}$ 与 e_{n-1}，则可定下 e_n，由此算出常数 α_n 和 β_n，最后可得到 s_n 的估计 $\hat{s}_n$，从而完成第 u 步计算。

2. 一维递归图像复原处理

在图像复原处理中，如果用一维卡尔曼滤波复原图像，首先必须用扫描的方式将二维平面信息转化为一维形式。这时观察到的图像为如下形式：

$$s(t) = s_I(t) + n(t) \tag{6-191}$$

式中 $s_I(t)$ 是理想图像，$n(t)$ 是加性噪声，$s(t)$ 是观测图像（或称数据样本）。由卡尔曼滤波原理，$s_I(t)$ 的第 k 个样本的递归估计量由下式得到：

$$\hat{s}_I(k) = \alpha_k \hat{s}_I(k-1) + \beta_k s(k) \tag{6-192}$$

这里 $\hat{s}_I(t)$ 是理想图像 $s(k)$ 在时刻 k 的估计，$\hat{s}_I(k-1)$ 是理想图像在时刻 $(k-1)$ 的估计，$s(k)$ 是新的观测值，α_k，β_k 则是系数。根据这一递归公式就可以在均方误差最小的准则下由含噪图像中恢复出原始图像。

以上是图像复原的卡尔曼滤波法的标量处理。另外，还可以采用矢量处理法。在这种方法中，同时扫描图像的几行，估计器的设计步骤与标量处理法相同。

3. 二维递归图像复原处理

图像复原的二维递归估计方法是由哈比比提出来的。直观地看，这种技术可以得到比标量处理及矢量处理更好的结果，因为二维处理可以利用图像相邻行的相关性，而在一维处理中却没有利用这一点。

哈比比提出的二维递归估计法是建立在这样的基础上，就是有一大类随机像场的自相关函数呈指数型，即可由下式来表示：

$$R(\tau_1,\tau_2) = \sigma_s^2 \exp\{-\alpha_1|\tau_1| - \alpha_2|\tau_2|\} \tag{6-193}$$

式中 τ_1 和 τ_2 是水平和垂直方向上的增量，α_1，α_2 是与图像有关的系数。具有这种自相关函数的离散随机场可用下面的平稳自回归源来表示：

$$\begin{aligned} x(k+1,l+1) = {} & \rho_1 x(k+1,l) + \rho_2 x(k,l+1) \\ & - \rho_1\rho_2 x(k,l) + \sqrt{(1-\rho_1^2)(1-\rho_2^2)}u(k,l) \end{aligned} \tag{6-194}$$

式中 ρ_1 和 ρ_2 分别是水平和垂直方向上相邻点的相关系数，$u(k,l)$ 是和图像元素有相同方差的不相关的随机场。正像式(6-194)所描述的那样，$x(k,l)$ 是一个自回归场，这个场中的每一点都可以用其相邻近的点来预测，其均方误差由下式表示：

$$\begin{aligned} e^2 &= (1-\rho_1^2)(1-\rho_2^2)E\{u^2(k,l)\} \\ &= (1-\rho_1^2)(1-\rho_2^2)\sigma_s^2 \end{aligned} \tag{6-195}$$

这里 e^2 是 $x(k,l)$ 随机性的度量，σ_s^2 是 $x(k,l)$ 和 $u(k,l)$ 的公共方差。

对离散随机像场 $x(k,l)$ 的估计可用递推线性滤波（二维卡尔曼滤波）得到。假设图像是被白噪声污染了的，而且信号和噪声的均值及方差都已知的情况下，与随机场和观测有关的动态模型为

$$x(k+1,l+1) = \rho_1 x(k+1,l) + \rho_2 x(k,l+1)$$

$$- \rho_1\rho_2 x(k,l) + \sqrt{(1-\rho_1^2)(1-\rho_2^2)}u(k,l) \tag{6-196}$$

$$y(k,l) = Cx(k,l) + Dv(k,l) \tag{6-197}$$

这里 $y(k,l)$是退化信号，$v(k,l)$是加性噪声，并且 $k=1,2,\cdots,M$；$l=1,2,\cdots,N$。式(6-196)表示，一个随机场 $x(k,l)$可以由一组不相关的随机变量 $u(k,l)$产生。如果 $u(k,l)$是可观测的，用上述模型和一组初始值就可以产生 $x(k,l)$。然而，在实践中仅能观测到含噪信号 $y(k,l)$，也就是 $y(k,l)$被输入到递归滤波器中去。因为递归滤波器的输出应该有与原始图像相同的相关模型，所以其输出有：

$$\begin{aligned}\hat{x}(k+1,l+1) = &F_1(k,l)\hat{x}(k+1,l) + F_2(k,l)\hat{x}(k,l+1) \\ &+ F_3(k,l)\hat{x}(k,l) + F(k,l)y(k,l)\end{aligned} \tag{6-198}$$

这里 $\hat{x}(k,l)$是 $x(k,l)$的一个估计。显然，式(6-198)是由三个 $x(k,l)$的以前的估值及一个含噪的观测值形成的。式中初始值 $\hat{x}(1,l)$，$\hat{x}(k,1)$，$k=1,2,\cdots,M$；$l=1,2,\cdots,N$ 是已知的。

对于 $x(k+1,l+1)$的最优线性估计 $\hat{x}(k+1,l+1)$的一个必要条件是在每一点的估计误差对所有的先前遇到的数据点正交，即

$$\begin{aligned}&E\{[x(k+1,l+1) - \hat{x}(k+1,l+1)]y(i,j)\} = 0 \\ &(i,j) \in R(k+1,l+1)\end{aligned} \tag{6-199}$$

这里 $R(k+1,l+1)$代表参数空间中左边和上边的一组点。把式(6-196)和式(6-198)代入式(6-199)中，得到：

$$\begin{aligned}E\Big\{&[\rho_1 - F_1(k,l)]x(k+1,l) + [\rho_2 - F_2(k,l)]x(k,l+1) \\ &- [\rho_1\rho_2 + CF(k,l) - F_3(k,l)]x(k,l) + \sqrt{(1-\rho_1^2)(1-\rho_2^2)}u(k,l) \\ &- DF(k,l)v(k,l)y(i,j)\Big\} = 0\end{aligned} \tag{6-200}$$

因为

$$E\{v(k,l)y(i,j)\} = 0$$

$$E\{u(k,l)y(i,j)\} = 0$$

对所有的(i,j)将有：

$$\begin{cases}F_1(k,l) = \rho_1 \\ F_2(k,l) = \rho_2 \\ F_3(k,l) = [\rho_1\rho_2 + CF(k,l)]\end{cases} \tag{6-201}$$

将式(6-201)代入式(6-198)有：

$$\begin{aligned}\hat{x}(k+1,l+1) = &\rho_1\hat{x}(k+1,l) + \rho_2\hat{x}(k,l+1) + [\rho_1\rho_2 + CF(k,l)]\hat{x}(k,l) \\ &+ F(k,l)y(k,l)\end{aligned} \tag{6-202}$$

式中参数 ρ_1,ρ_2,C 为已知，剩下的问题是确定 $F(k,l)$。

令

$$\tilde{x}(k,l) = x(k,l) - \hat{x}(k,l) \tag{6-203}$$

$$\tilde{y}(k,l) = y(k,l) - \hat{y}(k,l) \tag{6-204}$$

则有

$$E\{\tilde{x}(k+1,l+1)y(i,j)\} = 0 \tag{6-205}$$

由于$(i,j)\in R(k,l)$，令(6-205)中的$(i,j)=(k,l)$，并将式(6-203)和式(6-204)代入式(6-205)，利用式(6-196)、式(6-197)和式(6-198)可得到：

$$F(k,l)=\frac{C[\rho_1P_{10}(k,l)+\rho_2P_{01}(k,l)+\rho_1\rho_2P_{00}(k,l)]}{C^2P_{00}(k,l)+\sigma_N^2D^2} \tag{6-206}$$

$$P_{ij}(k,l)=E\{\tilde{x}(k,l)\tilde{x}(k+i,l+j)\} \tag{6-207}$$

σ_N^2是噪声的方差。$P_{ij}(k,l)$可由如下关系得到：

$$\begin{aligned}P_{i+1,j+1}(k,l)=&\rho_1P_{i+1,j}(k,l)+\rho_2P_{i,j+1}(k,l)\\&-[\rho_1\rho_2+CF(k+2,l+j)]P_{i,j}(k,l)\end{aligned} \tag{6-208}$$

在式(6-206)中求$F(k,l)$所必需的$P_{10}(k,l)$、$P_{01}(k,l)$，$P_{00}(k,l)$，在式(6-208)中可用$P_{ij}(k,l)$的先前点来确定。

由式(6-198)定义的二维递归估计只限于对指数自相关函数描述的图像作复原处理。

以上我们讨论了递归复原技术的基本概念和方法。它们是对加性白噪声污染了的图像进行递归线性估计。应该看到，二维统计递归滤波理论还处在开始阶段，还需解决某些理论上的问题。现在只是对特定的问题作特定的解决，尚没有一般的通用模型。

在本章介绍的图像复原处理中，应注意到所有的模型都以减小某种误差为准则，对视觉标准来说它未必是最佳的。总之，在这个领域中还有很多工作要做。

附录7　模糊图像恢复处理程序实例

```
//本程序对真彩色由水平(或垂直)运动造成的模糊图像的恢复处理。
图像尺寸:192×192×24bit
①图像参数初始化：BOOL CAntifilterDlg::OnInitDialog();
②图像数据读取：void CAntifilterDlg::OnOpenButton();
③图像退化处理：void CAntifilterDlg::OnFilterButton();
④图像逆滤波恢复：void CAntifilterDlg::OnRefilterButton();
⑤图像显示：void CAntifilterDlg::OnPaint();//
// antifilterDlg.h : header file//
#if !defined(AFX_ANTIFILTERDLG_H_5BD6C507_3B2B_11D2_8AE6_444553540000_INCLUDED_)
#define AFX_ANTIFILTERDLG_H_5BD6C507_3B2B_11D2_8AE6_444553540000_INCLUDED_
#if MSC_VER >= 1000
#pragma once
#endif // MSC_VER >  1000
/////////////////////////////////////////
// CAntifilterDlg dialog
class CAntifilterDlg : public CDialog
{
// Construction
public:
	CAntifilterDlg(CWnd * pParent = NULL);// standard constructor
////////////////////////
//My Code Start Here!
////////////////////////
```

```
	LPSTR m_lpDIBBit1;
	LPSTR m_lpDIBBit2;
	LPSTR m_lpDIBBit3;
	LPSTR m_lpBMPBit;
	BYTE * m_BMPBYTE;
	BYTE * m_DIBBYTE1;
	BYTE * m_DIBBYTE2;
	BYTE * m_DIBBYTE3;
	LPBITMAPINFOHEADER m_lpDIBHdr;
	BYTE RedArray[192][192];
	BYTE GreenArray[192][192];
	BYTE BlueArray[192][192];
	BYTE gRedArray[192][192];
	BYTE gGreenArray[192][192];
	BYTE gBlueArray[192][192];
	BYTE rRedArray[192][192];
	BYTE rGreenArray[192][192];
	BYTE rBlueArray[192][192];
	int a;
	int A;
	int T;
	int Bb;
////////////////////////////////////
//My Code End Here!
////////////////////////////////////
// Dialog Data
	//{{AFX_DATA(CAntifilterDlg)
	enum { IDD = IDD_ANTIFILTER_DIALOG };
	//}}AFX_DATA
	// ClassWizard generated virtual function overrides
	//{{AFX_VIRTUAL(CAntifilterDlg)
	protected:
	virtual void DoDataExchange(CDataExchange * pDX);	// DDX/DDV support
	//}}AFX_VIRTUAL
// Implementation
protected:
	HICON m_hIcon;
	// Generated message map functions
	//{{AFX_MSG(CAntifilterDlg)
	virtual BOOL OnInitDialog();
	afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
	afx_msg void OnPaint();
	afx_msg HCURSOR OnQueryDragIcon();
	afx_msg void OnOpenButton();
```

```
        afx_msg void OnFilterButton();
        afx_msg void OnRefilterButton();
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};
//{{AFX_INSERT_LOCATION}}
// Microsoft Developer Studio will insert additional declarations immediately before the previous line.
#endif
// ! defined(AFX_ANTIFILTERDLG_H_5BD6C507_3B2B_11D2_8AE6_444553540000_INCLUDED_)
// antifilterDlg.cpp : implementation file
//
#include "stdafx.h"
#include "antifilter.h"
#include "antifilterDlg.h"
#include "math.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////
// CAboutDlg dialog used for App About
class CAboutDlg : public CDialog
{
public:
        CAboutDlg();
// Dialog Data
        //{{AFX_DATA(CAboutDlg)
        enum { IDD = IDD_ABOUTBOX };
        //}}AFX_DATA
        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CAboutDlg)
        protected:
        virtual void DoDataExchange(CDataExchange * pDX); // DDX/DDV support
        //}}AFX_VIRTUAL
// Implementation
protected:
        //{{AFX_MSG(CAboutDlg)
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
        //{{AFX_DATA_INIT(CAboutDlg)
```

```
	//}}AFX_DATA_INIT
}
void CAboutDlg::DoDataExchange(CDataExchange * pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CAboutDlg)
	//}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
	//{{AFX_MSG_MAP(CAboutDlg)
		// No message handlers
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()
/////////////////////////////////////////////////
// CAntifilterDlg dialog
CAntifilterDlg::CAntifilterDlg(CWnd * pParent /* =NULL */)
	: CDialog(CAntifilterDlg::IDD, pParent)
{
	//{{AFX_DATA_INIT(CAntifilterDlg)
	//}}AFX_DATA_INIT
	// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
void CAntifilterDlg::DoDataExchange(CDataExchange * pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CAntifilterDlg)
	//}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CAntifilterDlg, CDialog)
	//{{AFX_MSG_MAP(CAntifilterDlg)
	ON_WM_SYSCOMMAND()
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_BN_CLICKED(IDC_OPEN_BUTTON, OnOpenButton)
	ON_BN_CLICKED(IDC_FILTER_BUTTON, OnFilterButton)
	ON_BN_CLICKED(IDC_REFILTER_BUTTON, OnRefilterButton)
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()
/////////////////////////////////////////////////
// CAntifilterDlg message handlers
BOOL CAntifilterDlg::OnInitDialog()
{
	CDialog::OnInitDialog();
```

```
	// Add "About..." menu item to system menu.
	// IDM_ABOUTBOX must be in the system command range.
	ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
	ASSERT(IDM_ABOUTBOX < 0xF000);
	CMenu* pSysMenu = GetSystemMenu(FALSE);
	if (pSysMenu != NULL)
	{
		CString strAboutMenu;
		strAboutMenu.LoadString(IDS_ABOUTBOX);
		if (!strAboutMenu.IsEmpty())
		{
			pSysMenu->AppendMenu(MF_SEPARATOR);
			pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
		}
	}
	// Set the icon for this dialog. The framework does this automatically
	// when the application's main window is not a dialog
	SetIcon(m_hIcon, TRUE);			// Set big icon
	SetIcon(m_hIcon, FALSE);		// Set small icon

	// TODO: Add extra initialization here
//////////////////
//My Code Start Here!
//////////////////
/* Function:
   Initialize the parametre!
*/
	m_lpDIBBit1 = NULL;
	m_lpDIBBit2 = NULL;
	m_lpDIBHdr = new BITMAPINFOHEADER;
	m_lpDIBHdr->biSize = (DWORD)sizeof(BITMAPINFOHEADER);
	m_lpDIBHdr->biWidth = 192L;
	m_lpDIBHdr->biHeight = 192L;
	m_lpDIBHdr->biPlanes = 1;
	m_lpDIBHdr->biBitCount = 24;
	m_lpDIBHdr->biCompression = BI_RGB ;
	m_lpDIBHdr->biSizeImage = 0L;
	m_lpDIBHdr->biXPelsPerMeter = 0L;
	m_lpDIBHdr->biYPelsPerMeter = 0L;
	m_lpDIBHdr->biClrUsed = 0L;
	m_lpDIBHdr->biClrImportant = 0L;
	T=100;
	A=122;
	Bb=10;
```

```
        a=8;
////////////////
//My Code Ends Here!
////////////////
        return TRUE; // return TRUE unless you set the focus to a control
}
void CAntifilterDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        if ((nID & 0xFFF0) == IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout.DoModal();
        }
        else
        {
                CDialog::OnSysCommand(nID, lParam);
        }
}
// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.
void CAntifilterDlg::OnPaint()
{
        if (IsIconic())
        {
                CPaintDC dc(this); // device context for painting
                SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);
                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics(SM_CXICON);
                int cyIcon = GetSystemMetrics(SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width() - cxIcon + 1) / 2;
                int y = (rect.Height() - cyIcon + 1) / 2;
                // Draw the icon
                dc.DrawIcon(x, y, m_hIcon);
        }
        else
        {
                CPaintDC dc(this);
//////////////////
//My Code Start Here!
//////////////////
//Display the original Image!
```

```
            ::SetDIBitsToDevice(dc.m_hDC,10,10,192,192,
                  0,0,0,192,m_lpDIBBit1,(LPBITMAPINFO)m_lpDIBHdr,
                  DIB_RGB_COLORS );
//Display the Filtering Image!
            ::SetDIBitsToDevice(dc.m_hDC,10,260,192,192,
                  0,0,0,192,m_lpDIBBit2,(LPBITMAPINFO)m_lpDIBHdr,
                  DIB_RGB_COLORS );
//Display the Recovery Image!
        ::SetDIBitsToDevice(dc.m_hDC,260,260,192,192,
                  0,0,0,192,m_lpDIBBit3,(LPBITMAPINFO)m_lpDIBHdr,
                  DIB_RGB_COLORS );
//////////////////
//My Code Ends Here!
//////////////////
            CDialog::OnPaint();
        }
}
HCURSOR CAntifilterDlg::OnQueryDragIcon()
{
        return (HCURSOR) m_hIcon;
}
void CAntifilterDlg::OnOpenButton()
{
/*
Function:
        read Bmp Data into the buff!
Input:
        No parametre!
Output:
        m_DIBBYTE1 the point of the original Data;
        m_DIBBYTE2 the point of the filtering Data;
         (Load the same data as m_DIBBYTE1 at first!)
        m_DIBBYTE3 the point of the recovery Data;
         (Load the same data as m_DIBBYTE1 at first!)
*/
////////////////////////////////
//My Codes Start Here;
////////////////////////////////
        CFileDialog dlg(TRUE);
        if(dlg.DoModal()!=IDOK)return;
        CString FileName;
        FileName=dlg.GetPathName();
        CFile file;
        if((! file.Open(FileName,CFile::modeRead|CFile::shareExclusive)))
```

```
	{
		MessageBox("Can't open the file!");
		return;
	}
	DWORD dwBitsSize;
	dwBitsSize = file.GetLength();
//Allocate the memory.
	HGLOBAL hBMP=::GlobalAlloc(GMEM_MOVEABLE|GMEM_ZEROINIT,dwBitsSize);
	if (hBMP == 0)
	{
		MessageBox("Can't allocate the memery of hBmp!");
		return ;
	}
	m_lpBMPBit = (LPSTR) ::GlobalLock( hBMP);
//read the data into the pointed memory.
	if(file.Read(m_lpBMPBit,dwBitsSize)! =dwBitsSize)
	{
		::GlobalUnlock(hBMP);
		::GlobalFree( hBMP);
		MessageBox("Can't read data into the buffer!");
		return ;
	}
      DWORD dwDIBSize;
	dwDIBSize=dwBitsSize-sizeof(BITMAPFILEHEADER)-sizeof(BITMAPINFOHEADER);
	HGLOBAL HDIB1=::GlobalAlloc(GMEM_MOVEABLE|GMEM_ZEROINIT,dwDIBSize);
	HGLOBAL HDIB2=::GlobalAlloc(GMEM_MOVEABLE|GMEM_ZEROINIT,dwDIBSize);
	HGLOBAL HDIB3=::GlobalAlloc(GMEM_MOVEABLE|GMEM_ZEROINIT,dwDIBSize);
	if(HDIB1==0)return;
	m_lpDIBBit1=(LPSTR)::GlobalLock(HDIB1);
	m_lpDIBBit2=(LPSTR)::GlobalLock(HDIB2);
	m_lpDIBBit3=(LPSTR)::GlobalLock(HDIB3);
//set the point to the pointed data in the byte form.
	m_BMPBYTE
(BYTE * )m_lpBMPBit+sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
	m_DIBBYTE1 = (BYTE * )m_lpDIBBit1;
	m_DIBBYTE2 = (BYTE * )m_lpDIBBit2;
	m_DIBBYTE3 = (BYTE * )m_lpDIBBit3;
	for(DWORD i = 0;i < dwDIBSize;i++)
	{
		m_DIBBYTE1[i] = m_BMPBYTE[i];
		m_DIBBYTE2[i] = m_BMPBYTE[i];
		m_DIBBYTE3[i] = m_BMPBYTE[i];
	}
	::GlobalUnlock(hBMP);
```

```
        ::GlobalUnlock(HDIB1);
        ::GlobalUnlock(HDIB2);
        ::GlobalUnlock(HDIB3);
Invalidate();
//////////////////////////////
//My Codes End Here.
//////////////////////////////
}
void CAntifilterDlg::OnFilterButton()
{
/*
Function:
        Image Process: Filter Image!
Input:
        Moving Parametre: a;
        (Here a=8 below!)
Output:
        m_DIBBYTE2 the point of the filtering Data;
*/
//////////////////////////////////
//My Codes Start here:
//////////////////////////////////
        a=8;
//allocate the BMP_data into the form of real picture_array.
        int irow,icollumn,i,j;
        double totalA;
        i=0;
        for(irow=0;irow<192;irow++)
                for(icollumn=0;icollumn<192;icollumn++)
                {
                        RedArray[irow][icollumn]=m_DIBBYTE1[i];
                        GreenArray[irow][icollumn]=m_DIBBYTE1[i+1];
                        BlueArray[irow][icollumn]=m_DIBBYTE1[i+2];
                        i=i+3;
                }
//Caculate the filter change
        //1. caculate the red filter change.
        for(irow=0;irow<192;irow++)
              for(icollumn=0;icollumn<192;icollumn++)
               {
                int ColorR,R;
                int t,m;
                double mR,p,q;
                ColorR=0;
```

```
        m=0;
        p=0;
        for(t=0;t<T;t++)
         {
            //caculate the p=x-a * t/1000;T=1000
            p=(float)icollumn-a * t/T;
            if(p>191||p<0)m=icollumn;
            else
            {
               q=p-floor((double)p);
               if(q>=0.5)m=(int)ceil((double)p);
               if(q<0.5) m=(int)floor((double)p);
               }
               ColorR=ColorR+(int)RedArray[irow][m];
            }
        mR=ColorR * 0.01;
        R=(int)ceil((double)mR);
        if(mR<0)R=0;
        if(mR>255)R=255;
        gRedArray[irow][icollumn]=(BYTE)R;
        }
//2. caculate the Green filter change with the same way of Red's.
for(irow=0;irow<192;irow++)
     for(icollumn=0;icollumn<192;icollumn++)
      {
        int ColorR,R;
        int t,m;
        double mR,p,q;
        ColorR=0;
        m=0;
        p=0;
        for(t=0;t<T;t++)
         {
          //caculate the p=x-a * t/1000;T=1000
          p=(float)icollumn-a * t/T;
          if(p>191||p<0)m=icollumn;
          else
             {
               q=p-floor((double)p);
               if(q>=0.5)m=(int)ceil((double)p);
               if(q<0.5) m=(int)floor((double)p);
             }
             ColorR=ColorR -(int)GreenArray[irow][m];
         }
```

```
            mR=ColorR * 0.01;
            R=(int)ceil((double)mR);
            if(mR<0)R=0;
            if(mR>255)R=255;
            gGreenArray[irow][icollumn]=(BYTE)R;
            }
        //3. caculate the Blue filter change with the same way as Red'.
        for(irow=0;irow<192;irow++)
            for(icollumn=0;icollumn<192;icollumn++)
            {
            int ColorR,R;
            int t,m;
            double mR,p,q;
            ColorR=0;
            m=0;
            p=0;
            for(t=0;t<T;t++)
             {
               //caculate the p=x-a * t/1000;T=1000
               p=(float)icollumn-a * t/T;
               if(p>191||p<0)m=icollumn;
               else
               {
                 q=p-floor((double)p);
                 if(q>=0.5)m=(int)ceil((double)p);
                 if(q<0.5) m=(int)floor((double)p);
               }
             ColorR=ColorR+(int)BlueArray[irow][m];
             }
            mR=ColorR * 0.01;
            R=(int)ceil((double)mR);
            if(mR<0)R=0;
            if(mR>255)R=255;
            gBlueArray[irow][icollumn]=(BYTE)R;
            }
    //allocate the picture array into the physicial memory form.
        j=0;
        totalA=0;
        for(irow=0;irow<192;irow++)
            for(icollumn=0;icollumn<192;icollumn++)
                {

                    m_DIBBYTE2[j]=gRedArray[irow][icollumn];
                    m_DIBBYTE2[j+1]=gGreenArray[irow][icollumn];
```

```
                    m_DIBBYTE2[j+2]=gBlueArray[irow][icollumn];
                    j=j+3;
                }
        Invalidate();
/////////////////
//My Code End Here!
/////////////////
}
void CAntifilterDlg::OnRefilterButton()
{
/*
Function:
      Image Process:Recovery Image!
Input:
      No Parametre.
Output:
      m_DIBBYTE3 the point of the recovery Data;
*/
///////////////////////
//My Code Starts Here:
///////////////////////
        int irow,icollumn,k,i,j;
        int p1,p2,q1,q2;
        double p,q,totalq;
        double R,G,B;
        int RR,gR;
        int m;
        float variation[123];
        double temp,tempR,tempG,tempB;
        int num,data;
        j=0;
        for(irow=0;irow<192;irow++)
                for(icollumn=0;icollumn<192;icollumn++)
                {
                gRedArray[irow][icollumn]=m_DIBBYTE2[j];
                        gGreenArray[irow][icollumn]=m_DIBBYTE2[j+1];
                        gBlueArray[irow][icollumn]=m_DIBBYTE2[j+2];
                        j=j+3;
                }
  for(num=0;num<123;num++)
  {
        temp=0;
        tempR=0;
        tempG=0;
```

```
        tempB=0;
        Bb=num;
//Caculate the recovery change
  //1.caculate the red recovery change.
      for(irow=0;irow<192;irow++)
        for(icollumn=0;icollumn<192;icollumn++)
                {
                        m=icollumn/a;
                        p1=icollumn-m*a;
                        if(p1==0)
                        {
                                p2=p1+1;
                         RR=(int)gRedArray[irow][p1];
                              gR=(int)gRedArray[irow][p2];
                         p=m*(gR-RR)*100;//0.01;
                        }
                        else
                        {
                        p2=p1-1;
                        RR=(int)gRedArray[irow][p1];
                        gR=(int)gRedArray[irow][p2];
                     p=m*(RR-gR)*100;
                        }
                        totalq=0;
                        q=0;
                        for(k=0;k<=m;k++)
                        {
                                q1=icollumn-a*k;
                                if(q1==0)
                                {
                                        q2=q1+1;
                        RR=(int)gRedArray[irow][q1];
                                gR=(int)gRedArray[irow][q2];
                      q=(gR-RR)*100;
                                }
                                else
                                {
                                q2=q1-1;
                                if(q2<0){q2=icollumn;MessageBox("Error");}
                                RR=(int)gRedArray[irow][q1];
                                gR=(int)gRedArray[irow][q2];
                           q=(RR-gR)*100;
                                }
                                totalq=totalq+q;
```

```
            }
            double totalA;
            int nn,n;
            totalA=0;
            for(n=0;n<=m;n++)
            {
                nn=icollumn+n*a;
                if(nn>191)nn=191;
                totalA=totalA+RedArray[irow][nn];
            }
            if (m==0) m=1;
            A=(int)floor(totalA/m);
            R=A-p+totalq;
            if(R<0)R=0;
            if(R>255)R=255;
            int b;
            b=(int)(gRedArray[irow][icollumn]);
            if(R<(b-Bb))
            {
            R=gRedArray[irow][icollumn];

            }
        else
            {
              R=(int)floor((double)R);
            }
          if(R>(b+B))
            {
              R=gRedArray[irow][icollumn];

            }
        else
            {
               R=(int)floor((double)R);
            }
            data=(int)(RedArray[irow][icollumn]);
            tempR=tempR+(R-data)*(R-data);
            rRedArray[irow][icollumn]=(BYTE)R;

        }
  //2. caculate the Green recovery change.
  for(irow=0;irow<192;irow++)
    for(icollumn=0;icollumn<192;icollumn++)
        {
```

```
m=icollumn/a;
p1=icollumn-m * a;
if(p1==0)
{
    p2=p1+1;
  RR=(int)gGreenArray[irow][p1];
    gR=(int)gGreenArray[irow][p2];
    p-m * (gR-RR) * 100;
}
else
{
p2=p1-1;
RR=(int)gGreenArray[irow][p1];
gR=(int)gGreenArray[irow][p2];
p=m * (RR-gR) * 100;
}
totalq=0;
q=0;
for(k=0;k<=m;k++)
{
    q1=icollumn-a * k;
    if(q1==0)
    {
        q2=q1+1;
RR=(int)gGreenArray[irow][q1];
    gR=(int)gGreenArray[irow][q2];
  q=(gR-RR) * 100;
    }
    else
    {
    q2=q1-1;
    if(q2<0){q2=icollumn;MessageBox("Error");}
    RR=(int)gGreenArray[irow][q1];
    gR=(int)gGreenArray[irow][q2];
  q=(RR-gR) * 100;
    }
    totalq=totalq+q;
}
double totalA;
int nn,n;
totalA=0;
for(n=0;n<=m;n++)
{
    nn=icollumn+n * a;
```

```
                if(nn>191)nn=191;
                totalA=totalA+GreenArray[irow][nn];
            }
            if (m==0) m=1;
            A=(int)floor(totalA/m);
            R=A-p+totalq;
            if(R<0)R=0;
            if(R>255)R=255;
            int b;
            b=(int)(gGreenArray[irow][icollumn]);
           if(R<(b-Bb))
            {
            R=gGreenArray[irow][icollumn];
                G=R;
                B=R;
            }
        else
            {
              R=(int)floor((double)R);
            }
         if(R>(b+B))
           {
            R=gGreenArray[irow][icollumn];
            }
        else
            {
             R=(int)floor((double)R);
            }
            data=(int)(GreenArray[irow][icollumn]);
            tempG=tempG+(R-data)*(R-data);
            rGreenArray[irow][icollumn]=(BYTE)R;
        }
//3.caculate the Blue recovery change.
  for(irow=0;irow<192;irow++)
    for(icollumn=0;icollumn<192;icollumn++)
        {
            m=icollumn/a;
            p1=icollumn-m*a;
            if(p1==0)
            {
                p2=p1+1;
             RR=(int)gBlueArray[irow][p1];
               gR=(int)gBlueArray[irow][p2];
             p=m*(gR-RR)*100;
```

```
}
else
{
p2=p1-1;
RR=(int)gBlueArray[irow][p1];
gR=(int)gBlucArray[irow][p2];
p=m*(RR-gR)*100;
}
totalq=0;
q=0;
for(k=0;k<=m;k++)
{
    q1=icollumn-a*k;
    if(q1==0)
    {
        q2=q1+1;
RR=(int)gBlueArray[irow][q1];
    gR=(int)gBlueArray[irow][q2];
q=(gR-RR)*100;
    }
    else
    {
    q2=q1-1;
    if(q2<0){q2=icollumn;MessageBox("Error");}
    RR=(int)gBlueArray[irow][q1];
    gR=(int)gBlueArray[irow][q2];
q=(RR-gR)*100;
    }
    totalq=totalq+q;
}
double totalA;
int nn,n;
totalA=0;
for(n=0;n<=m;n++)
{
    nn=icollumn-n*a;
    if(nn>191)nn=191;
    totalA=totalA+BlueArray[irow][nn];
}
if (m==0) m=1;
A=(int)floor(totalA/m);
R=A-p+totalq;
if(R<0)R=0;
if(R>255)R=255;
```

```
                int b;
                b=(int)(gBlueArray[irow][icollumn]);
                if(R<(b-Bb))
                {
                 R=gBlueArray[irow][icollumn];
                }
        else
                {
                 R=(int)floor((double)R);
                }
           if(R>(b+B))
               {
                R=gBlueArray[irow][icollumn];
               }
        else
               {
                R=(int)floor((double)R);
               }
        data=(int)(BlueArray[irow][icollumn]);
                tempG=tempG+(R-data) * (R-data);
                rBlueArray[irow][icollumn]=(BYTE)R;
        }
    temp=tempR+tempG+tempB;
    variation[num]=(float)(temp);
  }
//allocate the picture array into the physicial memory form.
    i=0;
    for(irow=0;irow<192;irow++)
        for(icollumn=0;icollumn<192;icollumn++){
                m_DIBBYTE3[i]=rRedArray[irow][icollumn];
                m_DIBBYTE3[i+1]=rGreenArray[irow][icollumn];
                m_DIBBYTE3[i+2]=rBlueArray[irow][icollumn];
                i=i+3;
        }
    Invalidate();
///////////////
//My Code Ends Here!
///////////////
}
```

附录 8　中值和均值滤波图像恢复处理程序实例

```
// PreProcessDoc.cpp : implementation of the CPreProcessDoc class(主程序)
//
```

```
#include "stdafx.h"
#include "PreProcess.h"
#include "PreProcessDoc.h"
#include "CDib.h"
#include "DlgFilter.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = _FILE_;
#endif
/////////////////////////////////////////////////////////////////////////////
// CPreProcessDoc
IMPLEMENT_DYNCREATE(CPreProcessDoc, CDocument)
BEGIN_MESSAGE_MAP(CPreProcessDoc, CDocument)
	//{{AFX_MSG_MAP(CPreProcessDoc)
	ON_COMMAND(ID_FILE_OPEN, OnFileOpen)
	ON_COMMAND(ID_PREPROCESS_FILTER, OnPreprocessFilter)
	ON_COMMAND(ID_MEANFILTER, OnMeanfilter)
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()
/////////////////////////////////////////////////////////////////////////////
// CPreProcessDoc construction/destruction
CPreProcessDoc::CPreProcessDoc()
{
	// TODO: add one-time construction code here
	//for(int i = 0;i<Row;i++)
	// for(int j = 0;j<Col;j++)
	// {
	//		m_inputimage[i][j] = RGB(255,255,255);
	//		m_outputimage[i][j] = RGB(255,255,255);
	// }
}
CPreProcessDoc::~CPreProcessDoc()
{
}
BOOL CPreProcessDoc::OnNewDocument()
{
	if (! CDocument::OnNewDocument())
		return FALSE;
	// TODO: add reinitialization code here
	// (SDI documents will reuse this document)
	return TRUE;
}
/////////////////////////////////////////////////////////////////////////////
```

```
// CPreProcessDoc serialization
void CPreProcessDoc::Serialize(CArchive & ar)
{
        if (ar.IsStoring())
        {
                // TODO: add storing code here
        }
        else
        {
                // TODO: add loading code here
        }
}
/////////////////////////////////////////////////
// CPreProcessDoc diagnostics
#ifdef _DEBUG
void CPreProcessDoc::AssertValid() const
{
        CDocument::AssertValid();
}
void CPreProcessDoc::Dump(CDumpContext & dc) const
{
        CDocument::Dump(dc);
}
#endif //_DEBUG
/////////////////////////////////////////////////////
// CPreProcessDoc commands
void CPreProcessDoc::OnFileOpen()
{
        // TODO: Add your command handler code here
        CFileDialog fileDialog(TRUE, "bmp", "*.bmp");
        // Display the Open dialog box.
        int result = fileDialog.DoModal();
        // If the user exited the dialog box
        // via the OK button...
        if (result == IDOK)
        {
          // Get the selected path and file name.
          CString string = fileDialog.GetPathName();
          // Construct a new CDib object.
          m_pDib = new CDib(string);
          // Check that the CDib object was created
          // okay.
          // If there was an error, the pBmInfo pointer
          // will be 0.
```

```
        LPBITMAPINFO pBmInfo = m_pDib->GetDibInfoPtr();
        // If the CDib object was not constructed
        // properly, delete it.
        if (! pBmInfo)
            DeleteContents();
        // Otherwise, set the document's title to
        // the DIB's path and file name.
        else
            {
            SetTitle(string);
                Transfer();
            }
    }
    // Notify the view object that it has new data
    // to display.
    UpdateAllViews(0);
}
void CPreProcessDoc::Median(int n)
{
    int y,x,yy,xx,n2,nn,chuo,chg,m,medi,madom,mado[1000];
    if(n<3||n%2 ! = 1) return;
    n2 = (n-1)/2;
    nn = n * n;
    chuo = (nn-1)/2;
    for(y=0;y<Row;y++)
    {
        for (x=0;x<Col;x++)
        {
            m_outputimage[y][x] = 0;
        }
    }
    for(y= n2;y<Row-n2;y++)
    {
        for(x= n2; x<Col-n2;x++)
        {
            m=0;
            for(yy = y-n2;yy<=y-n2;yy++)
                for(xx = x-n2;xx<x+n2;xx++)
                {
                    mado[m] = m_inputimage[yy][xx];
                    m++;
                }
                do
                {
```

```
                    chg = 0;
                    for(m=0;m<nn-1;m++)
                    {
                        if(mado[m]<mado[m+1])
                        {
                            madom=mado[m];
                            mado[m] = mado[m+1];
                            mado[m+1] = madom;
                            chg = 1;
                        }
                    }
                }
                while(chg == 1);
                medi = mado[chuo];
                m_outputimage[y][x] = medi;
                //display the out bitmap;
                }
        }
}
void CPreProcessDoc::Transfer()
{
    if(m_pDib)
        for(UINT i=0;i<m_pDib->GetDibWidth();i++)
            for(UINT j=0;j<m_pDib->GetDibHeight();j++)
                m_inputimage[i][j] = m_pDib->GetBitData(i,j);
}
void CPreProcessDoc::OnPreprocessFilter()
{
    // TODO: Add your command handler code here
    CDlgFilter dlg;
    if(dlg.DoModal()== IDOK)
    {
        if(m_pDib)
            Median(dlg.m_windowsize);
    }
    UpdateAllViews(0);
}
void CPreProcessDoc::OnMeanfilter()
{
    CDlgFilter dlg;
    if(dlg.DoModal()== IDOK)
    {
        if(m_pDib)
            Mean(dlg.m_windowsize);
```

```
        }
        UpdateAllViews(0);

}
void CPreProcessDoc::Mean(int n)
{
        int y,x,yy,xx,n2,sum,tmp2;
        unsigned char tmp1;
        if (n<3||n%2! =1) return;
        n2=(n-1)/2;
        for (y=0;y<Row;y++)
        {
                for (x=0;x<Col;x++)
                {
                        m_outputimage[y][x]=0;
                }
        }
        for (y=n2;y<Row-n2;y++)
        {
                for (x=n2;x<Col-n2;x++)
                {
                        sum=0;
                        for (yy=y-n2;yy<=y+n2;yy++)
                                for (xx=x-n2;xx<=x+n2;xx++){
                                        tmp1=(unsigned char)m_inputimage[yy][xx];
                                 sum+=(int)tmp1;
                                }
                                tmp2=(int)((float)sum/(n*n)+0.5);
                                tmp1=(unsigned char)tmp2;
                        m_outputimage[y][x]=(int)tmp1+(int)(tmp1<<8)-(int)(tmp1<<16);
                }
        }

        for (y=0;y<Row;y++)
        {
                for (x=0;x<Col;x++)
                {
                        //m_outputimage[y][x]=m_inputimage[y][x];
                // TRACE(" %3d ",(unsigned char)(m_inputimage[y][x]));
                }
        // TRACE("\n");
        }
}
// PreProcessDoc.h : interface of the CPreProcessDoc class
```

```
//
///////////////////////////////////////////////////
#if ! defined(AFX_PREPROCESSDOC_H_2467A96B_3CEF_11D3_A7D7_444553540000_INCLUDED_)
#define AFX_PREPROCESSDOC_H_2467A96B_3CEF_11D3_A7D7_444553540000_INCLUDED_
#include "CDib.h" // Added by ClassView
#if_MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000
#define Row 512
#define Col 512
class CDib;
class CPreProcessDoc : public CDocument
{
protected: // create from serialization only
        CPreProcessDoc();
        DECLARE_DYNCREATE(CPreProcessDoc)
// Attributes
public:
// Operations
public:
// Overrides
        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CPreProcessDoc)
        public:
        virtual BOOL OnNewDocument();
        virtual void Serialize(CArchive& ar);
        //}}AFX_VIRTUAL
// Implementation
public:
        void Transfer();
        COLORREF m_outputimage[Row][Col];
        COLORREF m_inputimage[Row][Col];
        void Median(int n);
        void Mean(int n);
        CDib * m_pDib;
        virtual ~CPreProcessDoc();
#ifdef _DEBUG
        virtual void AssertValid() const;
        virtual void Dump(CDumpContext& dc) const;
#endif
protected:
// Generated message map functions
protected:
        //{{AFX_MSG(CPreProcessDoc)
```

```
        afx_msg void OnFileOpen();
        afx_msg void OnPreprocessFilter();
        afx_msg void OnMeanfilter();
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};
/////////////////////////////////////////////////////
//{{AFX_INSERT_LOCATION}}
// Microsoft Developer Studio will insert additional declarations immediately before the previous line.
#endif
//! defined(AFX_PREPROCESSDOC_H_2467A96B_3CEF_11D3_A7D7_444553540000_INCLUDED_)
```

思 考 题

1. 试述图像退化的基本模型，并画出框图。

2. 什么是线性、空间不便的系统，试写出其表达式。

3. 试刻划出连续退化模型，何为点扩散函数？

4. 试写出离散退化模型。

5. 什么是约束复原？什么是非约束复原，在什么条件下进行选择？

6. 逆滤波复原的基本原理是什么？它的主要难点是什么？如何克服？

7. 试描述最小二乘方复原方法？

8. 如果不知道原始图像的功率谱，而只知道噪声的方差，采用何种方法复员图像为好？试述其原理。

9. 试述一维和二维中值滤波的基本原理。

10. 试描述窗口为 3 的一维加权的中值滤波原理及窗口为 3×3 的二维加权的中值滤波原理。

11. 试述均值滤波的基本原理。

12. 试对下图作 3×3 的中值滤波处理，并写出处理结果。

1	1	7	1	8	1	7	1	1
1	1	1	1	5	1	1	1	1
1	1	1	5	5	5	1	1	7
7	1	1	5	5	5	1	1	1
1	1	1	5	5	5	1	8	1
1	8	1	1	5	1	1	1	1
1	8	1	1	5	1	1	8	1
1	1	1	1	5	1	1	1	1
1	1	7	1	8	1	7	1	1

13. 对上图作 3×3 的均值滤波，并比较均值滤波与中值滤波的差异。

第7章　图像重建

7.1　概述

图像处理中一个重要研究分支是物体图像的重建，它被广泛应用于检测和观察中，而这种重建方法一般是根据物体的一些横截面部分的投影而进行的。在一些应用中，某个物体的内部结构图像的检测只能通过这种重建才不会有任何物理上的损伤。由于这种无损检测技术的显著优点，因此，它的适用面非常广泛，它在各个不同的应用领域中都显示出独特的重要性。例如：医疗放射学、核医学、电子显微、无线和雷达天文学、光显微和全息成像学及理论视觉等等领域都多有应用。

三维重建处理中研究的主要问题及各种不同的重建方案如图7-1中所示。

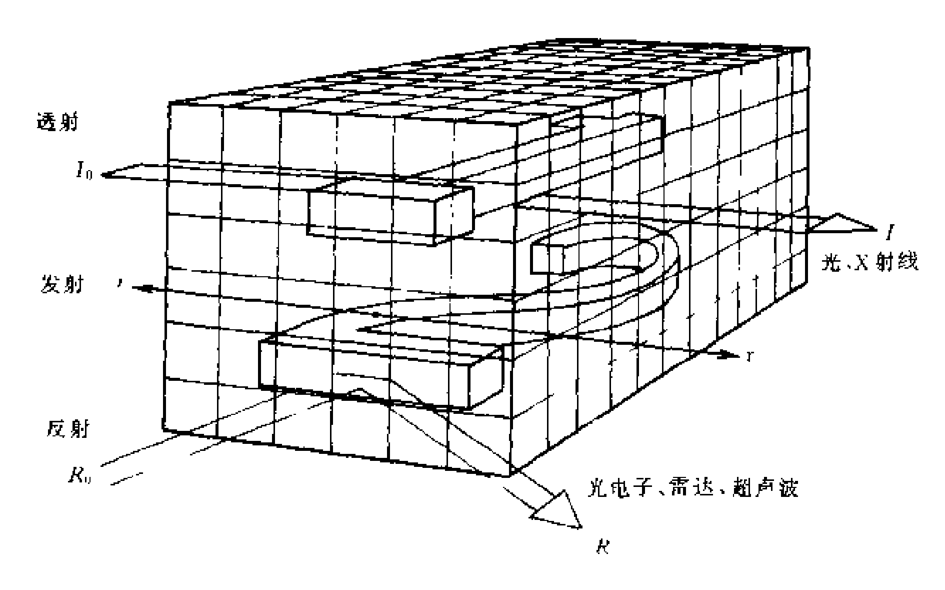


图7-1　图像重建的透射、反射、发射三种模式示意图

假设，两个嵌在内部的物体只能从外边观察，那么，采用什么检测手段才能达到这样的目的呢？当然，将物体切开是一种显而易见的解决方法。然而，在许多情况下这样做是不实际的，比如说，医疗检查、天文观测、工业中的无损检测、光传导中的测量等一些应用都不能采用这种破坏性方法。在目前的技术条件下，上述应用中的每一种情况，利用一些特殊的检测方法确定内部物体的特征，无论是在理论上还是在应用上都是有可能的。图7-1中示出了三种检测模型，即透射模型、发射模型和反射模型。透射模型建立于能量通过物体后有一部分能量会被吸收的基础之上，透射模型经常用在X射线、电子射线及光线和热辐射的情况下，这些都遵从一定的吸收法则。发射也可用来确定物体的位置，并且这种方法已经广泛用于正电子检测，它是通过在相反的方向分解散射的两束伽马射线来实现的。这两束射线的渡越时间可用来确定物体的位置。能量反射也可用来测定物体的表面特性，例如，光线、电子束、激光或做为能量源的

超声波等都可以用来进行这种测定。这三种模型是无损检测中常用的数据获取方法。这一章我们集中研究三维重建问题，正像绪论中谈到的那样，图像重建处理经多年研究已取得巨大进展，也产生了许多有效的算法，如：代数法、迭代法、傅里叶反投影法、卷积反投影法等，其中以卷积反投影法运用最为广泛，因为它的运算量小、速度快。近年来，由于与计算机图形学相结合，把多个二维图像合成三维图像，并加以光照模型和各种渲染技术，已能生成各种具有强烈真实感及纯净的高质量三维人造图像。这些方法融入了三维图形的主要算法，如：线框法、表面法、实体法、彩色分域法等等。此外，三维重建技术也是当今颇为热门的虚拟现实和科学可视化技术的基础。

7.2 傅里叶变换重建

傅里叶变换是最简单的重建方法。一个三维（或二维）物体，它的二维（或一维）投影的傅里叶变换恰与此物体的傅里叶变换的主体部分相等，而傅里叶变换重建方法也正是以此为基础的。通过将投影进行旋转和傅里叶变换可以首先构造整个傅里叶变换的平面，然后只须再通过傅里叶反变换就可以得到重建后的物体。

1974 年由 Shepp 和 Logan 开发的傅里叶变换重建的原理如下：

令 $f(x,y)$ 代表一图像函数，则此二维函数的傅里叶变换为

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp[-\mathrm{j}2\pi(ux+vy)]\mathrm{d}x\mathrm{d}y \tag{7-1}$$

而图像在 x 轴上的投影为

$$g_y(x) = \int_{-\infty}^{+\infty} f(x,y)\mathrm{d}y \tag{7-2}$$

投影的一维傅氏变换为

$$G_y(u) = \int_{-\infty}^{+\infty} g_y(x)\exp(-2\pi ux)\mathrm{d}x = \int_{\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp(-\mathrm{j}2\pi ux)\mathrm{d}x\mathrm{d}y \tag{7-3}$$

由此式可见，它恰与二维傅氏变换的表达式一致。即：

$$F(u,0) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp(-\mathrm{j}2\pi ux)\ \mathrm{d}x\mathrm{d}y \tag{7-4}$$

现在假设将函数投影到一条经过旋转的直线上，该直线的旋转角度为 θ。如图 7-2 所示。

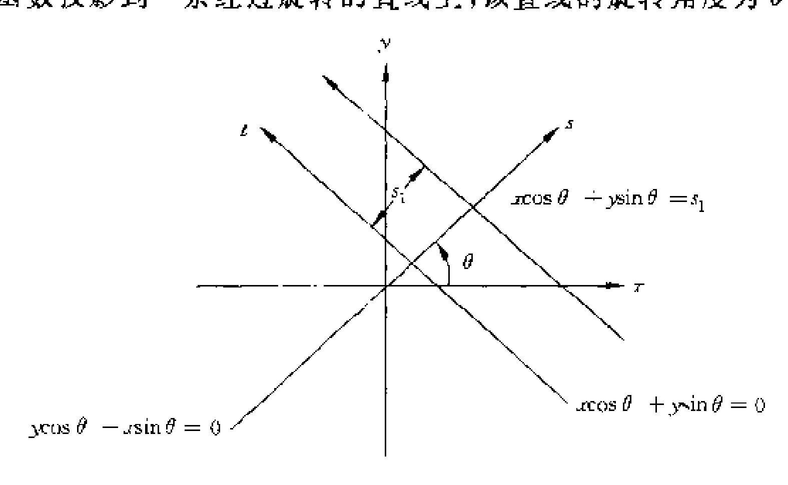


图 7-2　投影几何关系

首先，定义旋转坐标为

$$s = x\cos\theta + y\sin\theta \qquad t = -x\sin\theta + y\cos\theta \tag{7-5}$$

将函数投影的直线选为 x 轴。投影点通过对距离 t 轴为 s_1 处的一平行线进行函数积分，因此，该投影可如下表示：

$$g(s_1,\theta) = \int_{s_1} f(x,y)\mathrm{d}s \tag{7-6}$$

这里，积分路径是沿着直线 $s_1 = x\cos\theta + y\sin\theta$ 进行。此投影的一维傅氏变换为

$$G(r,\theta) = \int_{-\infty}^{+\infty} g(s_1,\theta)\exp(-\mathrm{j}2\pi r s_1)\mathrm{d}s_1 \tag{7-7}$$

展开后为

$$G(r,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp[-\mathrm{j}2\pi r(x\cos\theta + y\sin\theta)]\mathrm{d}x\mathrm{d}y \tag{7-8}$$

为使展开式与投影的二维傅里叶变换相等，把指数项做某种代换得到：

$$u = r\cos\theta \qquad v = r\sin\theta \tag{7-9}$$

因而，若点 (u,v) 在一条 θ 角一定而距原点距离为 r 的直线上，投影变换将与二维变换中的一直线有相同的傅氏变换，即 $F(u,v) = G(r,\theta)$。显然，若投影变换 $G(r,\theta)$ 中的所有 r 及 θ 值都是已知的，则图像的二维变换也是可以确定的。为得到图像函数，我们须进行反变换运算，即

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)\exp[\mathrm{j}2\pi(ux + vy)]\mathrm{d}x\mathrm{d}y \tag{7-10}$$

这就是重建技术的基础。

这些结论很容易推广到三维情形中。令 $f(x_1,x_2,x_3)$ 表示一物体，这里 f 可为实数或复数。它的三维傅氏变换由下式给出：

$$F(u_1,u_2,u_3) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x_1,x_2,x_3)\exp[-2\pi\mathrm{j}(u_1x_1 + u_2x_2 + u_3x_3)]\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x_3 \tag{7-11}$$

而变换的核心部分是

$$F(u_1,u_2,0) = \int^{+\infty}\int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty} f(x_1,x_2,x_3)\mathrm{d}x_3\right]\exp[-2\pi\mathrm{j}(u_1x_1 + u_2x_2)]\mathrm{d}x_1\mathrm{d}x_2 \tag{7-12}$$

通过定义，纵剖面或在 x_1,x_2 面上的投影是

$$f_3(x_1,x_2) = \int_{-\infty}^{+\infty} f(x_1,x_2,x_3)\mathrm{d}x_3 \tag{7-13}$$

注意到 $f_3(x_1,x_2)$ 的二维傅里叶变换正好等于上述三维变换的核心部分。这也说明如果投影在 x_1,x_2 在平面上旋转了 θ 角度，相应的傅里叶变换部分正好也将在变换域内的 u_1,u_2 平面内转过 θ 角。这样，投影可以采用不同的方向角 θ 插入到三维变换域中。建立一个傅里叶变换空间需要很多的投影。最后，通过傅里叶反变换重建图像 $f(x_1,x_2,x_3)$。既然在三维空间中的任意平面都可以被重建，那么，一个二维图像 $f(x_1,x_2)$ 的重建也不失一般性。所以，我们可重写二维投影方程，定出 θ 及投影平面 ρ：

$$g(\rho,\theta) = \int_S f(x,y)\mathrm{d}s \tag{7-14}$$

这里 $\mathrm{d}s$ 是光线几何路径中的微分长度。而傅里叶变换的结论由下面给出：

$$F(R,\theta) = \int_{-\infty}^{+\infty} g(\rho,\theta)\exp(-\mathrm{j}2\pi R\rho)\mathrm{d}\rho \tag{7-15}$$

和

$$f(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}F(u,v)\exp[\mathrm{j}2\pi(ux+vy)]\mathrm{d}u\mathrm{d}v \tag{7-16}$$

如图 7-3 所示。图中(a)是投影数据,(b)是傅里叶变换的组合。若已知无数的投影,从极坐标 $F(R,\theta)$ 中计算得到的投影变换推出在矩形平面 $F(u,v)$ 中的傅里叶变换并不困难。但是,若只有有限个投影是有效的,则可能需要在变换中插入一些数据。需要注意的是,虽然只须一维傅里叶变换的投影数据就可构成变换空间,但图像重建则需要二维反变换。由此,我们得出一个推论,即:三维图像不能在得到部分投影数据过程中局部地重建,而必须延迟到所有投影数据都获得之后才能重建。

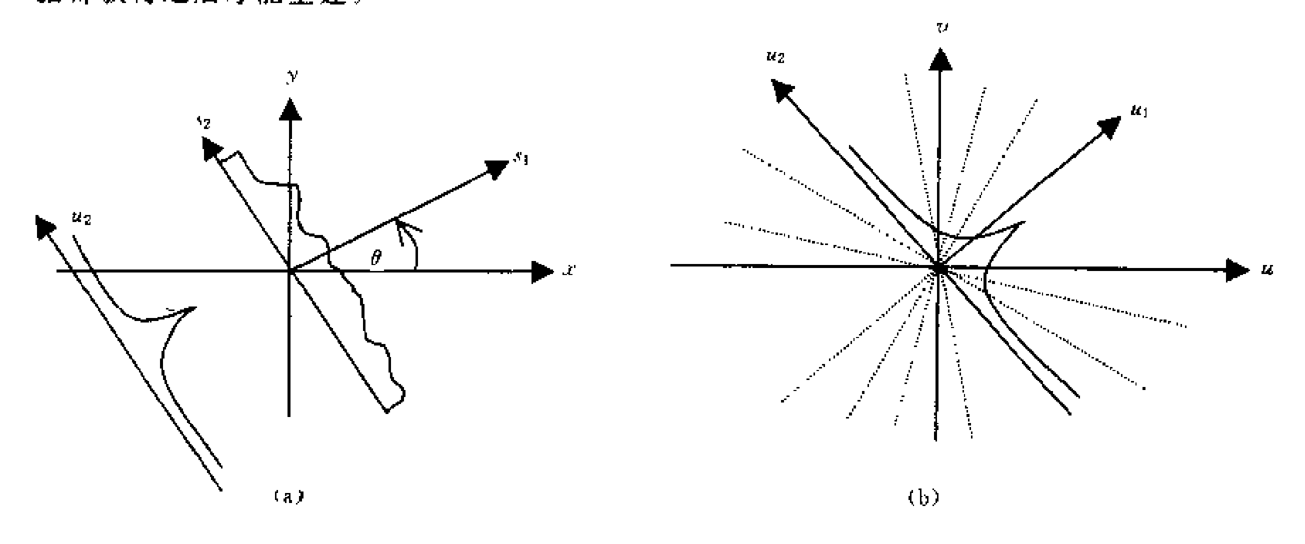


图 7-3　傅里叶变换的几何原理

7.3 卷积法重建

在讨论卷积法重建之前,先看一下图 7-4 中所示的极坐标中的傅里叶反变换表达式。

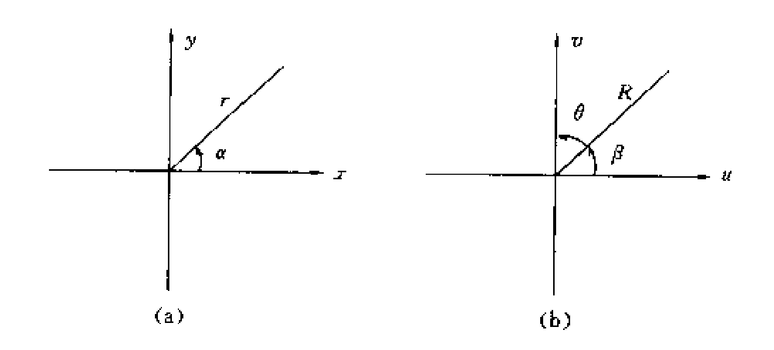


图 7-4　傅里叶变换的极坐标表示

图中(a)是空间域,(b)是变换域。其中笛卡儿坐标和极坐标的关系如下:

$$x=r\cos\alpha \qquad u=R\cos\beta=-R\sin\theta$$

$$y=r\sin\alpha \qquad v=R\sin\beta=R\cos\theta \tag{7-17}$$

$$f(r,\alpha)=\int_{0}^{2\pi}\int_{-\infty}^{+\infty}F(R,\theta)R\cdot\exp[\mathrm{j}2\pi Rr\sin(\alpha-\theta)]\mathrm{d}R\mathrm{d}\theta \tag{7-18}$$

由对称共轭特性可得到：

$$f(r,\alpha)=\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\int_{-\infty}^{+\infty}|R|F(R,\theta)\exp[\mathrm{j}2\pi Rr\sin(\alpha-\theta)]\mathrm{d}R\mathrm{d}\theta \tag{7-19}$$

令 $|R|\equiv R\mathrm{sgn}(R)\equiv-\mathrm{j}2\pi[\mathrm{j}\pi\mathrm{sgn}(R)]/2\pi^2$

这里，

$$\mathrm{sgn}(R)=\begin{cases}1 & R>0\\ 0 & R=0\\ -1 & R<0\end{cases}\qquad z=r\sin(\alpha-\theta)$$

则，

$$f(r,\alpha)=\frac{1}{2\pi^2}\int_{-\pi/2}^{\pi/2}[\mathrm{j}\pi\mathrm{sgn}(R)F(R,\theta)\exp(\mathrm{j}2\pi Rz)]\mathrm{d}R\mathrm{d}\theta \tag{7-20}$$

此表达式亦可写成下列形式：

$$f(r,\alpha)=\frac{1}{2\pi^2}\int_{-\pi/2}^{\pi/2}\left[\frac{\partial g(z,\theta)}{\partial z}\right]*\left\{\frac{1}{z}\right\}\mathrm{d}\theta \tag{7-21}$$

此处，* 代表卷积运算。此卷积表达示可直接写成：

$$f(r,\alpha)=\frac{1}{2\pi^2}\int_{-\pi/2}^{\pi/2}\int_{-\infty}^{+\infty}\left[\frac{\partial g(\rho,\theta)/\partial\rho}{r\sin(\alpha-\theta)-\rho}\right]\mathrm{d}\theta \tag{7-22}$$

这里，过 ρ 的积分可以解释为 Hilbert 在 $r=\sin(\alpha-\theta)$ 对 $g(\rho,\theta)$ 求偏导的变换式。这种解释的重要性在于：若取样值个数为有限的，则积分值为有限的，也就是收敛的。应注意到，前面所写的含有 $|R|$ 的积分表达式(7-19)不总是收敛的。

另外，这样求导也可推出一种很简便的图像重建方法。假定将投影数据 $g(\rho,\theta)$ 都存放于一等量矩形空间内，这种存放数据的方式称为 Layergram。对于一恒定 θ 值，我们可线性地滤出该投影数据，即可在频域内用 Rho 滤波器乘以 |R| 得出，也可以在空间域内通过一个滤波器冲激响应是 Rho 频率滤波器的反变换的投影数据卷积得出，表达式为

$$h_\theta(\rho)=\int_{-A/2}^{A/2}|R|\exp(\mathrm{j}2\pi R\rho)\mathrm{d}R \tag{7-23}$$

此处，积分上下限 A 是无限的，但在实际中一定为有限值。

这一处理就是所谓的 Rho Filtered Layergram 方法。为得到最终的重建图像，只需将 Rho Filtered Layergram 对 θ 在一特定 $\rho=r\cos(\alpha-\theta)$ 值作积分，即

$$f(r,\alpha)=\int_0^{\pi}g'(r\cos(\alpha-\theta),\theta)\mathrm{d}\theta \tag{7-24}$$

此处 $g'(\rho,\theta)=g(\rho,\theta)*h_\theta(\rho)$。

这个处理过程如图 7-5 所示，图中(a)是投影数据卷积，(b)是对于卷积的 Rho 滤波。

因为这一重建技术只需用到一维滤波和积分，因而在重建处理中具有极大吸引力。另外，该方法可以很容易产生与极坐标中的图像 $f(r,\alpha)$ 相对应的矩形值。

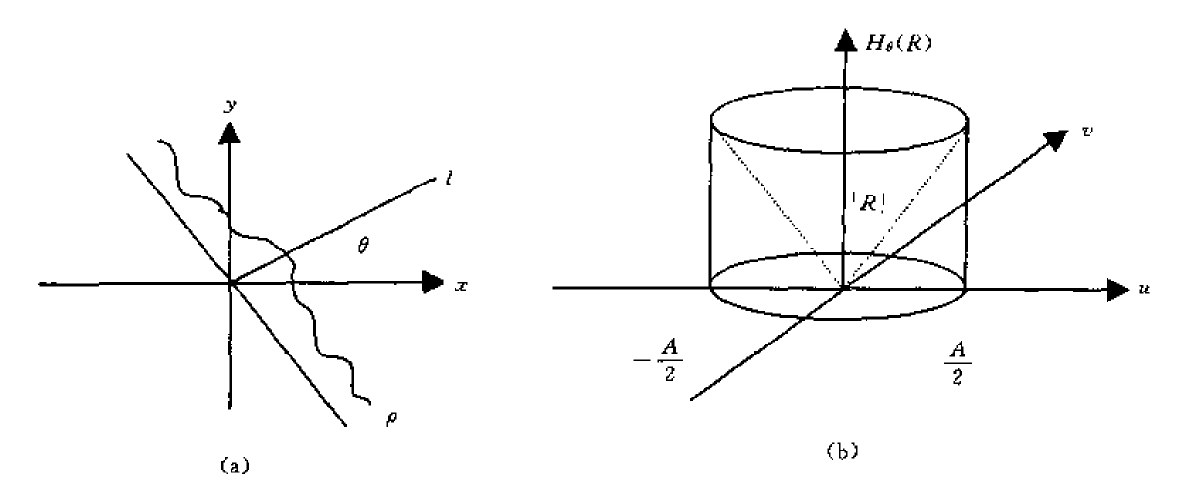


图 7.5 卷积技术图示

(a)卷积技术的几何表示.(b)Rho 滤波

7.4 代数重建方法

下面我们讨论重建的数字计算。由于数字传感器的动态范围不断增大，重建运算都用计算机来实现。基本的映射公式的数值计算需要如下几个步骤来实现。基本投影公式为

$$g(\rho,\theta) = \int_S f(x,y)\mathrm{d}s \tag{7-25}$$

首先，映射数据 ρ,θ 的取值必须为离散的。如：

$$g(\rho_m,\theta_n) \qquad m = 1,2,\cdots,M \quad n = 1,2,\cdots,N \tag{7-26}$$

其次，可以直接使用下面的积分式：

$$f(x,y) = \iint a(r,s)H(x,y,r,s)\mathrm{d}r\mathrm{d}s \tag{7-27}$$

如果积分是进行数字化计算，可以用级数来表示函数估值 $\hat{f}(x,y)$：

$$\hat{f}(x,y) = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} a_{kl}H_{kl}(x,y) \tag{7-28}$$

此处，系数 a_{kl} 及激励函数 $H_{kl}(x,y)$ 由重建选定的方法来决定。

级数估值应同时满足投影方程，即

$$g(\rho,\theta) = \int_S \hat{f}(x,y)\mathrm{d}s = \int_S f(x,y)\mathrm{d}s \tag{7-29}$$

或

$$g(\rho,\theta) = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} a_{kl}\int_S H_{kl}(x,y)\,\mathrm{d}s \tag{7-30}$$

系数 a_{kl} 需选择一定的值才可以满足上述条件。当然，由于变量取值是有限的，这种条件并不总能得到满足。最后，我们希望仅通过一系列离散的点来得到估计图像：

$$\hat{f}(x_i,y_j) \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,J \tag{7-31}$$

此外，当作此估值时应将量化效应考虑进去。当得到一组矩阵式后，则可以在某些特定条件下得到预期的估计。为了说明这些数字化重建中遇到的问题及解决方案，我们再讨论一下傅里叶变换的方法。

对于大多数重建问题，一个合理的假设是待重建的图像是空间有限的，即 $f(x,y)$ 在定义的矩形范围外 $\left\{|x|\leqslant\frac{1}{2}L_x,|y|\leqslant\frac{1}{2}L_y\right\}$ 均为零。在这样的假设下，直接应用抽样理论，首先可以推出 $f(x,y)$ 的周期延拓函数 $f_P(x,y)$，它在感兴趣的矩形周期内与 $f(x,y)$ 等效，并且在平面内周期性地重复。由于 $f_P(x,y)$ 具有周期性，我们可以将它写成二维傅里叶级数形式如下：

$$f_P(x,y)=\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}G_{mn}\exp\left[\mathrm{j}2\pi\left(\frac{mx}{L_x}+\frac{ny}{L_y}\right)\right] \tag{7-32}$$

这里傅里叶级数系数由下式给出：

$$G_{mn}=\frac{1}{L_xL_y}\int_{-L_x/2}^{L_x/2}\int_{-L_y/2}^{L_y/2}f(x,y)\exp\left[-\mathrm{j}2\pi\left(\frac{mx}{L_x}+\frac{ny}{L_y}\right)\right]\mathrm{d}x\mathrm{d}y \tag{7-33}$$

此级数在预先确定了系数的情况下，可以被截短而得到一有限级数表达式。

现在，利用在证明抽样理论中曾经用过的方法，我们在变换域中用等间隔栅格上的函数插值方法表示 $f(x,y)$ 的傅里叶变换如下：

$$F(u,v)=L_xL_y\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}G_{mn}\mathrm{sinc}\left[L_x\left(\frac{m}{L_x}-u\right)\right]\mathrm{sinc}\left[L_y\left(\frac{n}{L_y}-v\right)\right] \tag{7-34}$$

这里系数 G_{mn} 等于在 $(u=m/L_x,v=n/L_y)$ 点的傅里叶变换抽样值，其中，$\mathrm{sinc}(x)=\frac{(\sin\pi x)}{\pi x}$。

在 m、n 取有限值的情况下，如果能确定这些系数，也即 $m=0,1,2,\cdots,M-1,n=0,1,2,\cdots,N-1$，则可以用有限傅里叶级数表达式来计算 $f(x,y)$ 在感兴趣范围内任意点的值。这样，在给出傅里叶变换在极坐标中计算出的投影点后，我们需要确定系数 G_{mn}。

$F(u_i,v_j)$ 表示成极坐标形式，

$$\begin{aligned}u_i&=R_i\cos\theta_i \qquad i=1,2,\cdots,I\\ v_j&=R_j\sin\theta_j \qquad j=1,2,\cdots,J\end{aligned} \tag{7-35}$$

现在可以写出用等值向量形式表示的有限插入公式：

$$\boldsymbol{F}=\boldsymbol{WG} \tag{7-36}$$

此处 $\boldsymbol{F}$ 是一组下标为 IJ 的矢量，$\boldsymbol{G}$ 是下标为 MN 的矢量。则 $\boldsymbol{W}$ 是按 $IJ\times MN$ 排列的加权矩阵。到现在为止，该问题已简化为与线性恢复相同形式的问题了。这样，可以写出逆矩阵。即：

$$\boldsymbol{G}=(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})^{-1}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{F} \tag{7-37}$$

此方程可能并不易于计算。转置矩阵的阶数有待确定，这一阶数等于频域的点数。例如，要确定变换域内一个 80×80 栅格上的点，若直接解决该问题，需要转置一个 6400×6400 的矩阵。由此，我们得出结论：傅里叶变换法的数字计算可以说是一种级数法。

另外一种数字法是基于 $f(x,y)$ 是带限函数这一假设的基础上的方法，也就是：

$$F(u,v)=0 \qquad |u|\geqslant\frac{1}{2l_x} \quad |v|\geqslant\frac{1}{2l_y}, \tag{7-38}$$

在这种情况下，$f(x,y)$ 可以用基函数内插的方法来表示：

$$f(x,y)=\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}f(ml_x,nl_y)\mathrm{sinc}\left(\frac{x-ml_x}{l_x}\right)\mathrm{sinc}\left(\frac{y-nl_y}{l_y}\right) \tag{7-39}$$

该式的截短形式可以提供一种估算 $\hat{f}(x,y)$ 的方法，即：

$$\hat{f}(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(ml_x,nl_y)\mathrm{sinc}\left(\frac{x-ml_x}{l_x}\right)\mathrm{sinc}\left(\frac{y-nl_y}{l_y}\right)\mathrm{d}s \tag{7-40}$$

沿直线上的积分可由下式给出：

$$\hat{g}(\rho,\theta)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(ml_x,nl_y)\int_S \mathrm{sinc}\left(\frac{x-ml_x}{l_x}\right)\mathrm{sinc}\left(\frac{y-nl_y}{l_y}\right)\mathrm{d}s \tag{7-41}$$

为了估算沿直线的积分，直线方程由下式给出：

$$y=ax+b \tag{7-42}$$

通过定义

$$x'=x-ml_x \qquad y'=y-nl_y \tag{7-43}$$

可以方便地在任何一个取样点重新定位原点。

对于一个给定的 ρ 和 θ，直线方程为

$$y'=ax'+b' \tag{7-44}$$

这里，$a'=\tan\theta$，$b'=\rho\sec\theta+ml_x\tan\theta-nl_y$

$$W_{mn}(\rho,\theta)=\int_{-\infty}^{+\infty}\mathrm{sinc}\left(\frac{x-ml_x}{l_x}\right)\mathrm{sinc}\left(\frac{x-nl_y}{l_y}\right)(1+a^2)^{1/2}\mathrm{d}x \tag{7-45}$$

为了对这个表达式进一步求值，得到加权函数如下式：

$$W_{mn}(\rho,\theta)=\begin{cases}(1+a^2)^{1/2}l_x\mathrm{sinc}\left(\dfrac{\rho c+ml_xa-nl_y}{l_y}\right) & 0\leqslant|a|\leqslant\dfrac{l_y}{l_x}\\ (1+a^2)^{1/2}\left(\dfrac{l_y}{|a|}\right)\mathrm{sinc}\left(\dfrac{\rho c+ml_xa-nl_y}{l_xa}\right) & \dfrac{l_y}{l_x}<|a|<+\infty\\ l_y\mathrm{sinc}\left(\dfrac{\rho+ml_x}{l_x}\right) & |a|=+\infty\end{cases} \tag{7-46}$$

这里，$a=\tan\theta$，$c=\sec\theta$。所以，我们得到了一系列代数表达式：

$$g(\rho,\theta)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}W_{mn}(\rho,\theta)f(ml_x,nl_y) \tag{7-47}$$

式中，已知投影值及加权函数是一组离散值，并且从中可以确定离散点的图像值。实际上，加权函数必须通过一有限宽的射线来计算，因而，这会使过程稍复杂一些。

下面说明级数方法中的最后部分，假定图像由一个矩阵表示，如图 7-6 所示，并且每一元素内的函数具有一致的取值，比如说 $\bar{f}(ml_x,nl_y)$，则任意点(x,y)的函数可表示如下：

$$\hat{f}(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\bar{f}(ml_x,nl_y)\mathrm{rect}\left(\frac{x-ml_x}{l_x}\right)\mathrm{rect}\left(\frac{y-nl_y}{l_y}\right) \tag{7-48}$$

式中：

$$\mathrm{rect}(x)=\begin{cases}1 & |x|\leqslant\dfrac{1}{2}\\ 0 & |x|>\dfrac{1}{2}\end{cases}$$

在模拟方式中，此函数可以沿一射线路径来积分，这个积分宽度可能是有限的情况下来确定一系列加权函数$W'_{mn}(\rho,\theta)$，以开发线性系统，即

$$g(\rho,\theta)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\bar{f}(ml_x,nl_y)W'_{mn}(\rho,\theta) \tag{7-49}$$

重建方程的数字解决方法可以使我们得到一线性方程系统，从而解决确定该图像的问题，对于这样一个已成熟的问题，这里不再详述，仅简单提几点。首先，由于任意角度投影值的总和等于图像函数的积分，

$$\int g(\rho,\theta)\mathrm{d}\rho=\iint f(x,y)\mathrm{d}x\mathrm{d}y \tag{7-50}$$

可以容易地在系统中引入相关方程式。这样，就产生了一个单一系统方程。其次，这也使

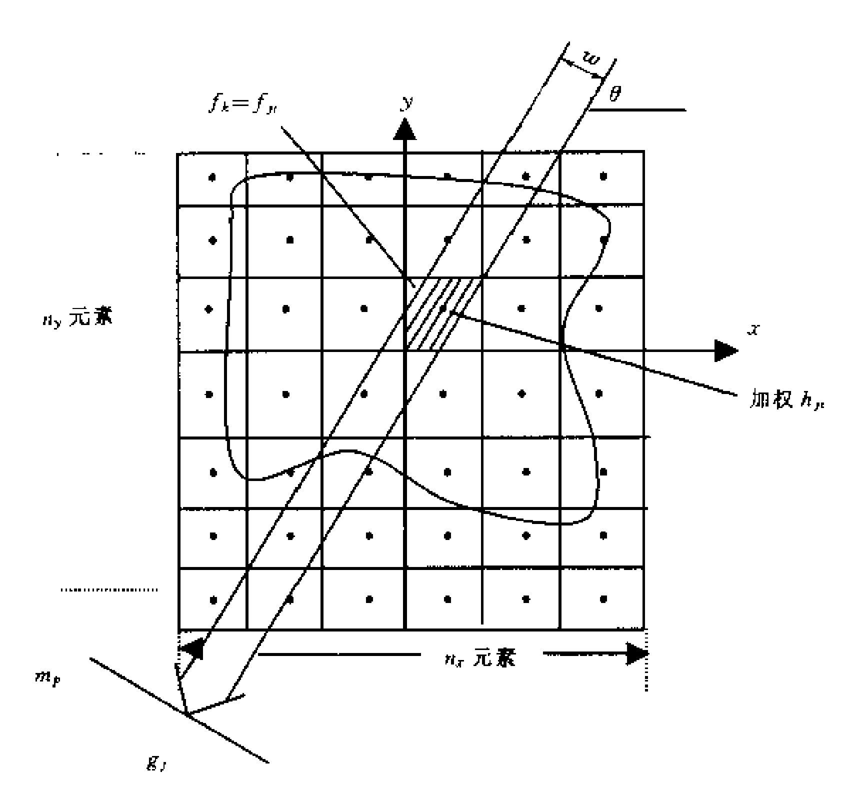


图 7-6　代数重建几何原理

引用如下一些客观标准成为可能。如最小均方误差，以及这组可行性方案中的一些优先约束因素。最后，我们可以使用直接、迭代或直接-迭代混合算法来使问题近似成为一系列线性方程表示。这样一来，将使近似算法变得很容易。

7.5　重建的优化问题

图像重建中的问题也可以通过选择一合理的准则函数来解决。此函数用来衡量真实图像与重建图像之间的差异，并且开发一种使此准则函数最小的解决方案。Kaskyap 和 Mittal 于 1975 年巧妙地将重建问题转变成最小化(函数)问题，目前已有多种基于该准则的代数解决方案，下面介绍其原理。

首先引入向量符号来表示重建投影。令 $\boldsymbol{f}$ 代表一图像向量，此向量通过将图像行向量 $\boldsymbol{f}_i$ 堆成一列向量而形成，即

$$\boldsymbol{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_n)^{\mathrm{T}} \tag{7-51}$$

如图 7-6 所示，向量的大小为 n^2。

其次，考虑到投影射线是以相对于水平角度 θ_k 入射，如图 7-7 所示，还应注意到 P 个这样的投影，其角度分别从 $\theta_1, \theta_2, \cdots, \theta_p$，令 $\boldsymbol{g}_k$ 是角度为 θ_k 时投影的向量值，显然，可以认为它有 n 个分量组成，

$$\boldsymbol{g}_k = (g_{k1}, g_{k2}, \cdots, g_{kn})^{\mathrm{T}}$$

将各角度的投影向量纵向排列，可得到 pn 个分量组成的向量 $\boldsymbol{g}$，

$$\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_p)^{\mathrm{T}} \tag{7-52}$$

如果投影值假定为下述图像值的线性组合，则

$$g_{kl} = \sum_{j \in D_{(k+1)n+l}} f_j \qquad k = 1,2,\cdots,p \qquad l = 1,2,\cdots,n \tag{7-53}$$

其中集合 D_j 由投影 g_j 组合的所有元素组成。如图 7-6 所示。注意到某个几何加权也能够计算出来。它是与射线宽度 w 和图 7-6 中所示的元素 f_{ij}的图形单元的交集部分有关的。为了简便起见，可不这样做。比较合适的方法是，如果一条入射角为 θ_k 的射线落在 f_{ij}单元内任一点，则总有一个元素可以用来组成投影值 g_{ij}。在这种假设下，有如下所述过程来得到元素的集合 D_j。

首先，在图像元素 f_{ij}取一单元(i,j)，此单元也可以称为单元 k，此处

$$k = (n-1)i + j \tag{7-54}$$

且令素元的中心在$\left(i-\frac{1}{2},j-\frac{1}{2}\right)$。

现在令 PQ 为图像主对角线 AB 的投影线，此时 OR 轴与水平轴夹角为 θ_k。如图 7-7 所示，则 PQ 长度为

$$\overline{PQ} = n\sqrt{2}\cos(45° - \theta_k) \tag{7-55}$$

将此长度平均分为 n 等分，$P_{k1},P_{k2},\cdots,P_{kn}$，如图 7-7 所示。将一元素分配给集合 D 的原则是：若单元(i,j)的中心投影落在相应的增量范围内，则认为此单元是投影的一部分。单元(i,j)在 OR 上的中心投影为

$$P_{ij} = \left[\left(i-\frac{1}{2}\right)^2 + \left(j-\frac{1}{2}\right)^2\right]\cos(q_{ij} - \theta_k) \tag{7-56}$$

这里，

$$q_{ij} = \arctan\left[\left(j-\frac{1}{2}\right)\Big/\left(i-\frac{1}{2}\right)\right] \tag{7-57}$$

如果单元(i,j)的投影 P_{ij}满足条件：

$$(l-1)\sqrt{2}\cos(45° - \theta_k) \leqslant P_{ij} \leqslant l\sqrt{2}\cos(45° - \theta_k) \tag{7-58}$$

则对一给定的 l 值，可以认为此点在集合 $D_{(K-1)N+1}$内。实质上，对于每个投影过程来说，每一图像元素的中心都被投影了，而上述条件则用以确定图像元素是否被投影了。

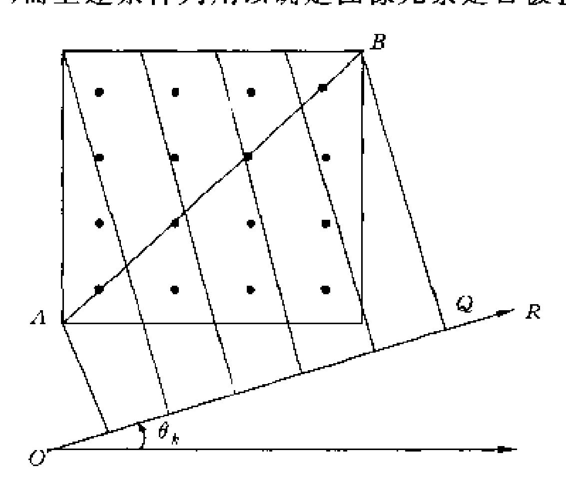


图 7-7 角度为 θ_k 的透影元素的赋值

下面讨论数学过程中存在的问题。投影方程的集合可以写成

$$\boldsymbol{g} = \boldsymbol{BF} \tag{7-59}$$

这里 B 是一个大小为 pn、n^2 个元素的二元矩阵：

$$B_{ij} = \begin{cases} 1 & j \in D_i \\ 0 & 其他 \end{cases} \tag{7-60}$$

注意到方程的解是：当且仅当 $p=n$ 时，有唯一解；当 $p<n$ 时，无解；当 $p>n$ 时，有多个解。现在图像重建问题已经简化为解线性方程组的问题。下一个要考虑的问题是准则函数的选择。

第一个准则 $J_1(\boldsymbol{f})$ 与局部是否平坦的或在一个局部一个元素与其邻点间的强度是否有差别有关。这时 $J_1(\boldsymbol{f})$ 叫做非均匀函数，其表达式如下：

$$J_1(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{f} \tag{7-61}$$

其中 $\boldsymbol{C}$ 是 8 邻域平滑矩阵。矩阵 $\boldsymbol{C}$ 是结构半正定的。注意到 $J_1(\boldsymbol{f})\geqslant 0$，对于均匀图像 $J_1(\boldsymbol{f})=0$。因此，需要使用约束条件来解决最小化问题。这可能会使不确定系统得不到唯一解。所以，考虑准则函数 $J_2(\boldsymbol{f})$：

$$J_2(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{f} + \alpha J_1(\boldsymbol{f}) \tag{7-62}$$

$J_2(\boldsymbol{f})$ 的第一项与图像的能量有关，也与样本方差 σ^2 有关。由于

$$\sigma^2 = E[(\boldsymbol{f}-\mu)^{\mathrm{T}}(\boldsymbol{f}-\mu)] = E(\boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}) - \mu^{\mathrm{T}}\mu \tag{7-63}$$

此处 $\mu=E(\boldsymbol{f})$，常量 α 可用实验方法来确定，可以获得最好的重建图像的那一个值便是常量 α。

现在，图像重建的问题就可归结为最小化 $J_2(\boldsymbol{f})=\frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}+\alpha\frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{f}$ 的问题了，这是由前述约束条件 $\boldsymbol{g}=\boldsymbol{BF}$ 推导出来的。而有约束的最小化问题可以通过引入一个 np 的拉格朗日(Lagrange)乘数向量 $\boldsymbol{\lambda}$ 加以解决。我们引入一新的准则函数：

$$J_3(\boldsymbol{f},\boldsymbol{\lambda}) = J_2(\boldsymbol{f}) + \boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{Bf}-\boldsymbol{g}) \tag{7-64}$$

它可被直接最小化，在考虑到 $\boldsymbol{f}$ 的情况下，为了最小化 $J_3(\boldsymbol{f})$，我们来计算偏导数：

$$\frac{\partial J_3}{\partial \boldsymbol{f}} = \boldsymbol{f} + \alpha\boldsymbol{C}\boldsymbol{f} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\lambda} \tag{7-65}$$

令偏导数为零，则得到：

$$(\boldsymbol{I}+\alpha\boldsymbol{C})\boldsymbol{f} = -\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\lambda} \tag{7-66}$$

由于 $\boldsymbol{C}$ 是一半正定矩阵，$\boldsymbol{I}+\alpha\boldsymbol{C}$ 是一非奇矩阵，因此，我们可以解出 $\boldsymbol{f}$：

$$\boldsymbol{f} = -(\boldsymbol{I}+\alpha\boldsymbol{C})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\lambda} \tag{7-67}$$

其中，$\boldsymbol{\lambda}$ 的值可以通过上式乘以 $\boldsymbol{B}$ 加以确定：

$$\boldsymbol{Bf} = -\boldsymbol{B}(\boldsymbol{I}+\alpha\boldsymbol{C})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\lambda} \tag{7-68}$$

这个结果与约束条件

$$\boldsymbol{g} = \boldsymbol{Bf} = -\boldsymbol{B}(\boldsymbol{I}+\alpha\boldsymbol{C})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\lambda} \tag{7-69}$$

相同。如果 $p<n$ 时，$\boldsymbol{B}$ 可能是非奇异的，就可能得到一个伪答案。如：

$$\boldsymbol{\lambda} = -[\boldsymbol{B}(\boldsymbol{I}+\alpha\boldsymbol{C})^{-1}\boldsymbol{B}^{\mathrm{T}}]^{\#}\boldsymbol{g} \tag{7-70}$$

这里，# 表示伪逆，对于重建图像 $\boldsymbol{f}$ 可以写成：

$$\boldsymbol{f} = \boldsymbol{Fg} \tag{7-71}$$

这里，

$$\boldsymbol{F} = (\boldsymbol{I}+\alpha\boldsymbol{C})^{-1}\boldsymbol{B}^{\mathrm{T}}[\boldsymbol{B}(\boldsymbol{I}+\alpha\boldsymbol{C})^{-1}\boldsymbol{B}^{\mathrm{T}}]^{\#} \tag{7-72}$$

注意到矩阵 $\boldsymbol{F}$ 仅与下列因素有关：参数 α，图像几何结构以及约束条件。所以 $\boldsymbol{F}$ 可以预先计算出来。

因为矩阵尺寸很大，并且需要采用逆矩阵的方法计算，则就计算方法来说，最优化重建方法并非是最简便的一种。Kashyap 和 Mittal 在 1975 曾提出了几种实用性很强的迭代方案。当然，最优化方案的可取之处在于其公式化。

7.6 图像重建中的滤波器设计

为说明滤波器设计中的问题，我们先复习最简单的解决方案即卷积算法的步骤，即：

1）收集投影数据 $g(\rho,\theta)$，并将数据存放于一矩形空间，即所谓的 Layergram。

2）对于一固定 θ 值，从 ρ 方向线性地过滤 Layergram，即用 $|R|$ 的逆变换对数据卷积：

$$h_\theta(\rho)=\int_{-A/2}^{A/2}|R|\exp(\mathrm{j}2\pi R\rho)\mathrm{d}R \tag{7-73}$$

3）在一特定 ρ 值，通过 Rho 滤波，对 Layergram 以 θ 为积分变量作积分，计算出反投影。即

$$\rho=r\cos(\alpha-\theta)$$

$$f(x,y)=f(\alpha,r)=\int_0^\pi g'(r\cos(\alpha-\theta),\theta)\mathrm{d}\theta \tag{7-74}$$

这里，$g'(\rho,\theta)=g(\rho,\theta)*h_\theta(\rho)$。

所以，图像重建主要包括三个步骤。第一步，数据采集。大体说就是用一矩形或扇形等几何形状的抽样线束，将投影数据收集并存放于 Layergram 中。第二步，滤波。滤波对重建图像的质量至关重要。第三步，反向投影。这是一个积分过程，此过程需要对每一个图像元素进行计算，因此，计算量很大。

图像重建的精确性在很大程度上依赖于滤波器 $h_\theta(\rho)$，目前已有一些性能良好的滤波器，它们都很接近 $|R|$ 的响应，但不是实际的而是理论上的理想响应。$|R|$ 滤波器的空间响应不是实际中的理想滤波器的响应，其原因在于它对无限的 A 不收敛，而事实上 A 一定为有限的，且对这一有限 A 会产生吉布斯(Gibbs)现象，此外，噪声的影响没有考虑进去。

几种滤波器的设计如下：

由 Ramachandran 和 Lakshiminarayanan(1971 年)定义的滤波器空间脉冲响应如下式：

$$h_1(0)=\frac{\pi}{2a^2}$$

$$h_1(ka)=\begin{cases}-\dfrac{2}{\pi k^2a^2} & k\text{ 为奇数}\\ 0 & k\text{ 为偶数}\end{cases} \tag{7-75}$$

这个滤波器当 $ka\leqslant\rho\leqslant(k+1)a$ 和 $\rho_k=ka$ 时用线性内插：

$$h(\rho)=h(ka)+\left[\frac{(\rho-\rho_k)}{a}\right]\{h[(k+1)a]-h[ka]\} \tag{7-76}$$

其频率响应函数是

$$H(\omega)=|\omega|\mathrm{sinc}^2\left(\frac{1}{2}\omega a\right)\qquad |\omega|\leqslant\frac{\pi}{a} \tag{7-77}$$

这里 $\mathrm{sinc}^2\left(\frac{1}{2}\omega a\right)$ 项来自于取样间的线性内插的结果。

Shapp 和 Logan(1974 年)用相同的线性内插改进了上面滤波器函数，即

$$h(ka)=-\frac{4}{\pi a^2(4k^2-1)}\qquad k=0,\pm1,\pm2,\cdots \tag{7-78}$$

其相应的频率响应为

$$H(\omega)=\left|\frac{2}{a}\sin\frac{\omega a}{2}\right|\mathrm{sinc}^2\left(\frac{\omega a}{2}\right) \tag{7-79}$$

为解决噪音的问题，Shapp 和 Logan 提出了一种噪音平滑滤波器：

$$h(\rho_k)=0.4h(\rho_k)+0.3h(\rho_k+a)+0.3h(\rho_k-a) \tag{7-80}$$

滤波器的频率响应为

$$\bar{H}(\omega)=0.4H(\omega)+0.6H(\omega)\cos(\omega)a \tag{7-81}$$

通用的 Reed-Kwoh 滤波器组的频率响应没有线性内插形式，即

$$H(\omega)=\begin{cases}a^{-1}|\omega|\exp(-\xi|\omega|^p), & |\omega|\leqslant\dfrac{\pi}{a}\\ a^{-1}\left|\dfrac{2\pi}{a-\omega}\right|\exp(-\xi\left|\dfrac{2\pi}{a}-\omega\right|^p) & \dfrac{\pi}{a}<|\omega|\leqslant\dfrac{2\pi}{a}\end{cases} \tag{7-82}$$

这里，ξ 被称为抑制因子，它可以决定截止频率，p 是一个滚降参数，它决定滤波器的尖锐性。由于滤波器的数字特性，$H(\omega)$ 被认为具有周期为 $2\pi/a$ 的周期性。线性内插通过因子 $\mathrm{sinc}^2\left(\frac{1}{2}\omega a\right)$ 来修改滤波器。

对于 $p=1$ 的情况，Reed-Kwoh 滤波器的冲激响应如下：

$$h(0)=(\pi\xi^2)^{-1}[1-\exp(-\beta\xi)(\beta\xi+1)] \tag{7-83}$$

并且

$$h(ka)=[\pi C_1^2(k)]^{-1}\{(-1)^{k+1}[\beta\xi C_1(k)+C_2(k)\exp(-\beta\xi)+C_2(k)]\} \tag{7-84}$$

对于 $k=\pm1,\pm2,\cdots$，有

$$\beta=\pi/a \qquad C_1(k)=\xi^2+a^2k^2 \qquad C_2(k)=\xi^2-a^2k^2$$

当 $P>1$，可用数字方式来得到冲激响应。

图 7-8 给出了几种滤波器频响特性的比较。

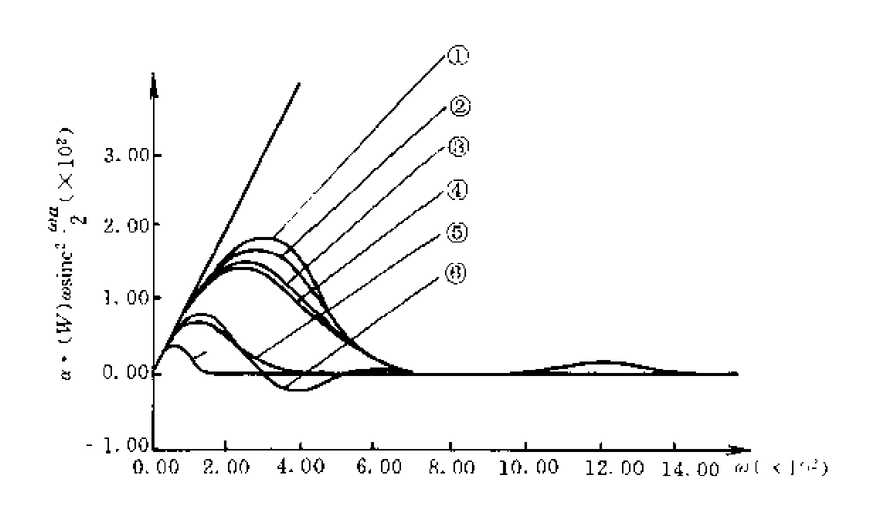


图 7-8　几种滤波器的响应

图 7-9 给出了滤波器对于模拟图像重建的作用，图中(a)是透视图，(b)是沿中心线的剖视图。两种滤波器的重建效果如图 7-10 所示。加入随机噪声后的进一步结果如图 7-11 所示。

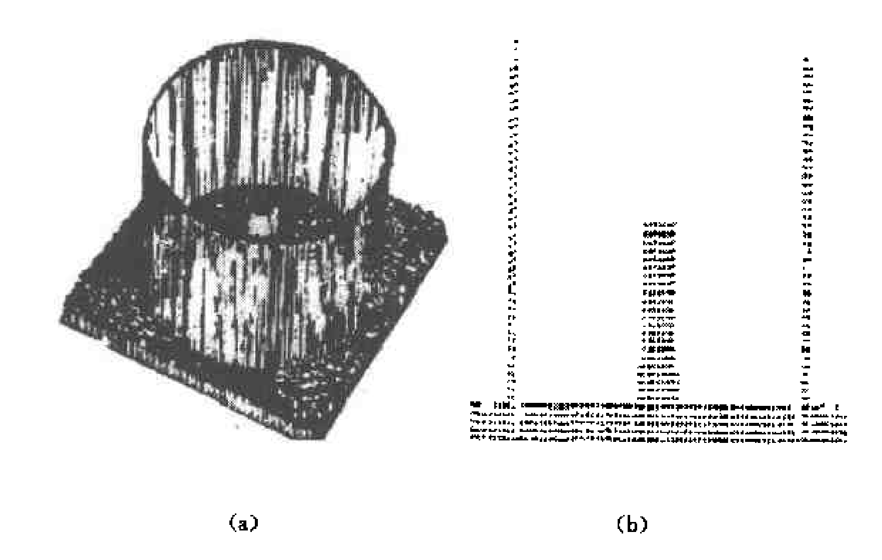


图 7-9 滤波器对仿真的作用

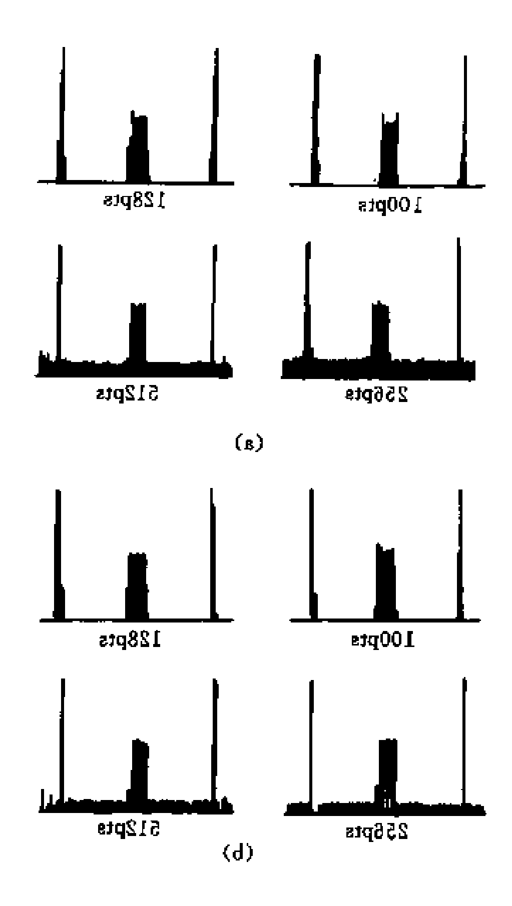


图 7-10 滤波效果(取自 Kwoh，1977)

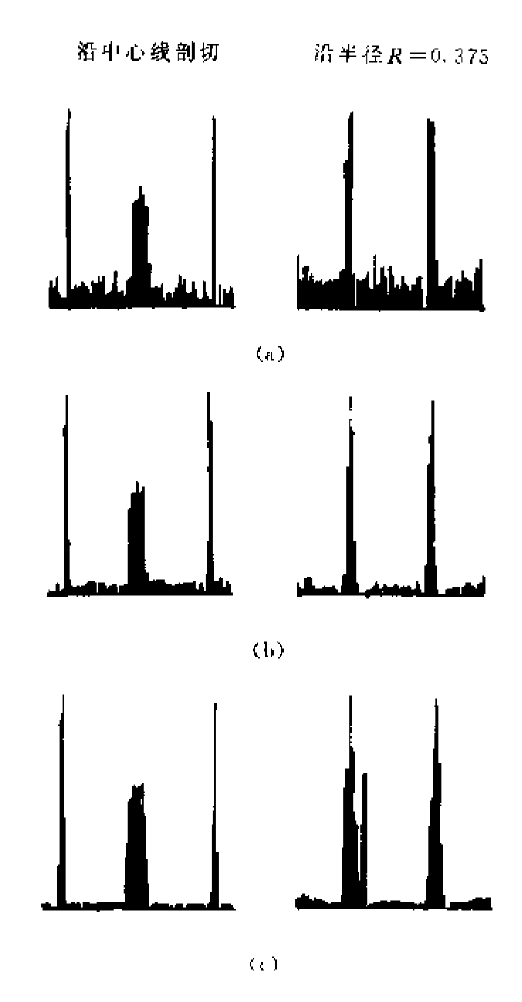


图 7-11　带有噪声的重建效果

7.7　重建图像的显示

图像重建的目的是对目标进行测量和观察，因此，重建图像中大量信息的直观显示是图像重建的任务之一。在有些应用中，如 X 射线图像的应用尤为重要。因为，我们得到的由一系列切片重建的三维矩阵中包含着大量瞬时信息，这些信息有的已超出了人的视觉范围，因此，人只能观察某些物体的表面特性。早期，常用的三维实体显示装置是用时间序列描述第三维信息，即用二维显示方法显示三维附加信息。采用这种方法的主要问题是单个切片的总信息不能在一幅图像中显示，而是需要一个图像的序列。这种显示方法的直观性是很差的。

7.7.1　重建图像的显示

在重建图像的显示中，首先要考虑图像的信息密度。如果一幅图像是 $N\times N$ 的矩阵，每一个像素包含 2^M 种可能的灰度，图像的总比特数为

$$T = N^2 M \tag{7-85}$$

要求图像显示的数目为

$$L = 2^T \tag{7-86}$$

例如：如果 $N=160$、$M=10$，则 $T=327680$、$L\approx10^{[illegible]}$。这样一来，每幅图像像素包含的最大信息为

$$H = \log 2^M = M \tag{7-87}$$

所以，具有1024级灰度的图像每像素可包含10比特的信息量。由于像素之间的相关性，实际的信息量将比这一最大信息量小得多。我们可以用计算每一像素的水平直方图的方法估计在一幅图像中的一阶熵，即

$$H = -\sum_{i=1}^{M} P_i \log_2 P_i \tag{7-88}$$

此外，我们还要考虑到分辨率 N 和每像素比特数之间并不是线性关系，然而，某些心理视觉资料表明对于相同的图像质量，N 与 M 之间的关系必须加以修正。同时，在重建图像的显示方法中必须考虑人的视觉系统对灰度范围和精确度的限制。尽管定量描述有些困难，但实验表明，在最好的观察条件下，人类仅能分辨几十种灰度、几千种不同的颜色和几秒的弧度，而大多数情况下视觉条件都难于达到最佳条件，因此，人眼能分辨的灰度级和颜色都是有限的。

就现代技术而言，在重建图像的显示中，可得到的空间分辨率比人眼的分辨率大得多。所以，在重建图像显示中可采用开窗的方法选择灰度区域来改善观察效果。

7.7.2 单色显示

三维重建图像的单色显示有如下几种装置，如：飞点扫描、CRT、机械微光图像密度计或打印机输出硬拷贝等。实际应用中阴极射线管(CRT)是典型的输出设备。在图像显示中的线性、量化、开窗口和增强(如平滑、锐化、高通滤波)处理是提高显示质量的必要技术。

线性处理是首先应考虑的预处理技术。给定一幅数字重建图像，数据和显示器灰度的关系具有非线性特性，为了获得数据与灰度之间的线性关系，必须考虑视觉条件和人的视觉系统。图7-12示出了两种通常的观察条件，一种是直接从CRT(阴极摄像管)得到观察图像，另一种是从摄影照片中得到观察图像。由于CRT和HD胶片的非线性线特性，同样的数字图像在两种条件下显示的灰度是非线性的。此外，如果人的视觉系统特性也考虑在内的话，线性关系的数字图像与实际感觉的灰度也是非线性的。这就说明如果没有校正步骤在两种观察条件下都不可能得到最佳质量。

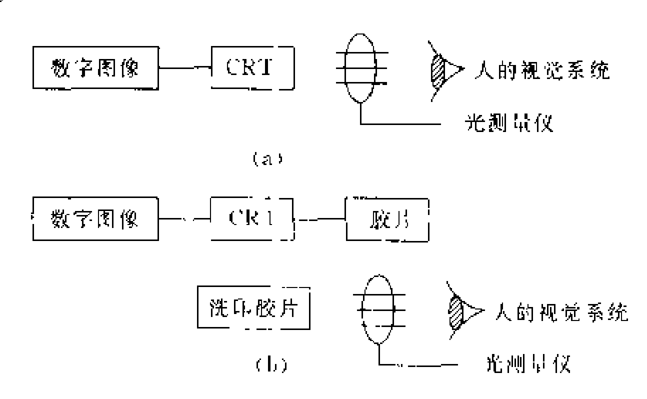


图7-12 观察条件

(a)直接CRT观察，(b)图片观察

在图7-12(a)中CRT的观察条件下，一给定点的发光强度 I 与电压 U 的关系可近似为

$$I = U^{\gamma} \tag{7-89}$$

这里 γ 与对比度有关。这一关系类似于照相胶片的"gamma"特性。如果电压值与图像数 N 成比例的话，则

$$U = kN \tag{7-90}$$

那么，发光强度与图像数 N 成指数关系：

$$I = (kN)^{\gamma} \tag{7-91}$$

另一方面，如果图像用负幂数来表示，则图像可用 N' 表示：

$$N' = N^{\gamma^{-1}} \tag{7-92}$$

于是，在发光强度和图像之间就可以得到一个线性关系：

$$I = (kN')^{\gamma} = (kN^{\gamma^{-1}})^{\gamma} = k^{\gamma}N \tag{7-93}$$

对于一个给定的 CRT，γ 的值很容易测得。

由于胶片的非线性，观察图片的情况就更为复杂。

众所周知，一般都用黑白反转的底片印制照片，它是由一覆盖细微银粒的感光乳剂的底片制成的。这样，随着底片吸收反射光强的变化，控制感光乳剂银粒的沉积数量而形成图像。银粒密度动态范围在 50：1 到 100：1 范围内。当一幅照片把图像信息传递给观察者时，观察者对照明光度的细微变化并不敏感，这个现象是由视觉系统的适应能力引起的。对光学图像复制特性的研究表明，尽管绝对亮度影响感知质量，但在任何给定的复制环境下其对物理限制的质量标准很大程度上依赖于它对相对亮度比例的复制能力，这个事实在直观上是令人满意的。应注意的是像素亮度比是场景反射的一个特性，该反射相对于一个均匀亮度的绝对强度是不变的。

把黑白乳剂暴露在光线下就产生了一个用它的光学密度描述的亮度吸收层，其中 D 定义为对入射光传送比例的对数。当其他参数固定时，光学密度和曝光强度 I 的理论关系式可表示为

$$D = \gamma_1 \lg(It) \tag{7-94}$$

其中 t 为曝光时间。这个公式就是在摄影界很有名的 Hurter-Driffield 或 D-$\lg E$ 曲线。实际上，照相器材在它们的动态范围内并不遵循这个理想化规律。参数 γ_1 描述了对应于原始负片的正片乳剂的对比度，反之亦然。因为没曝光的乳剂和它的基片并不是完全透明的，所以，在以上的等式中加入了一个修正值 D_0，上式就变为

$$D = D_0 + \gamma_1 \lg(It) \tag{7-95}$$

来自照片的反射光 I 和入射光 I_0 的关系式为

$$I' = I_0 10^{-D}(It) \quad 或者 \quad D = -\lg(I/I_0) \tag{7-96}$$

由 CRT 产生的光强的数量关系为

$$I = N^{\gamma_1} \tag{7-97}$$

当用该光强来使照相胶片曝光时，产生的密度 D 为

$$D = \gamma_2 \lg It = \gamma_2 \lg(N^{\gamma_1}t) \tag{7-98}$$

如果强度为 I_R 的光是从亮度为 I_0 的光由密度为 D 的胶片反射出来的，则

$$D = \lg(I_R/I_0) \tag{7-99}$$

同样可得到强度值：

$$I_R = I_0 N^{\gamma_1\gamma_2} \tag{7-100}$$

这样，就得到了一个由 CRT 的 gamma 值所确定的数字图像所决定的反射光强与胶片光强的关系式，如果

$$N' = N^{(\gamma_1\gamma_2)^{-1}} \tag{7-101}$$

那么，

$$I_R = I_0[N^{(\gamma_1\gamma_2)^{-1}}]^{\gamma_1\gamma_2} \quad 或者 \quad I_R = I_0 N \tag{7-102}$$

于是，只要得到胶片的 gamma 值 γ_2，对 CRT 和照片就可以作矫正了。整个的失真曲线和补偿曲线如图 7-13 所示。

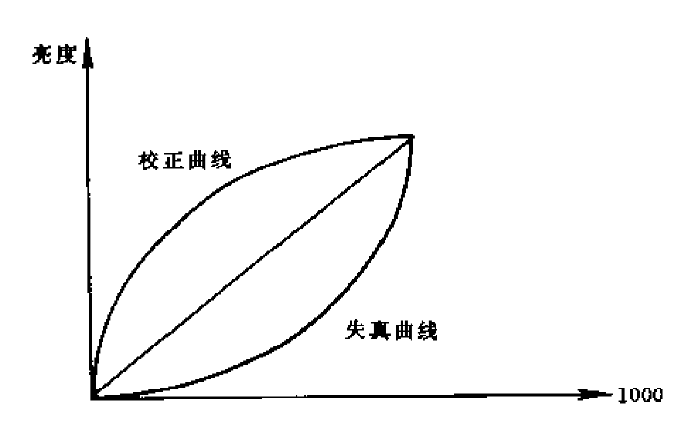


图 7-13 CRT 补偿及胶片 gamma 校正

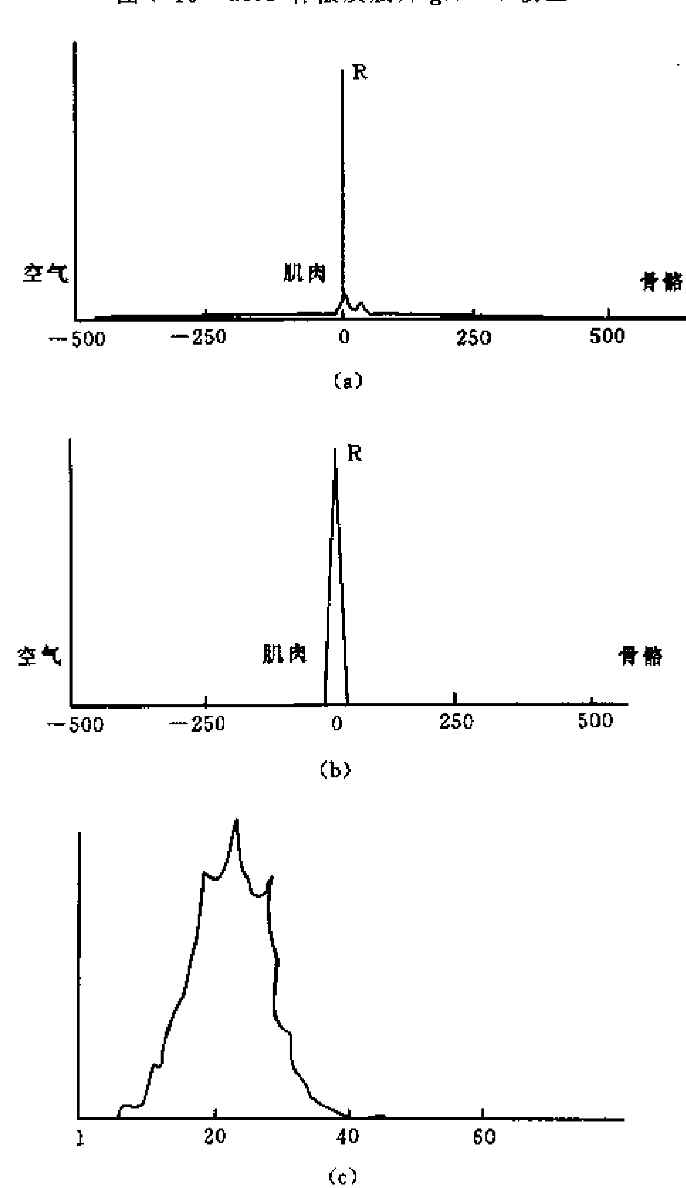


图 7-14 (a)头部扫描 EMI 直方图、(b)脑内部计算决定的直方图，(c)直方图表尺为 1～60 时，肿瘤灰度位于峰值区 20 处

在重建图像显示中，窗口技术的采用也是很重要的。在重建图像中量化的级数可以作到0.1%(1000级)，但是图像观察装置只允许观察50级，这时，就必须采用窗口技术去观察图像。为使用方便起见，有必要考虑使用交互窗口，利用人机交互的方法改变窗口位置，以便观察不同灰度范围的图像。正如图7-14所示的那样，直方图是在各个灰度级上像素数的积累值。图中大部分的像素值都是聚集在中心区域。窗口中心和宽度可以从原始的1000级量化级的计算机断层图像(CT)的直方图得到。由图可见，大部分有用信息都集中在稍高于中心区域的地方。例如，如果在峰值两边延伸大约20级灰度的宽度，这样，一个中心为20，整个宽度为40的窗口区域就被选出来描述原始计算机断层图，结果如图7-15(a)所示。

为了验证选取的窗口是否合适，以下的方法经实验证明非常有效。在用直方图法选出窗口的中心和宽度后，原始的计算机断层摄影图就被重新量化和显示。



(a)



(b)

图7-15 开窗法显示效果

(a)中心为20，宽度为40的效果；(b)中心为20，宽度为20的显示效果

在结合点上，可疑的对象和鉴定出来的肿瘤区域就会标识出来。一个交互式游标就移到这个可疑处并读出其灰度值，根据相邻像素灰度值的不同可以得到新宽度的另一个值。新窗口中心和宽度的灰度值为20。图7-15(b)显示一幅重新量化的图像。总而言之，直方图法和交互式修改使重建图像(特别是在低对比度时)能更好地显示出来。

7.7.3 重建对象的显示

重建信息三维矩阵的显示本身就是一个复杂的问题。近年来，经科技工作者的不懈努力已有多种方法可供选择。其中最基本的方法是显示密度信息和表面信息。在大多数应用中，由重建算法所得到的密度信息可以直接在收集了投影数据的几何薄片上显示。第三维信息可以用一组二维图像简单描述显示出来。

由于薄片的厚度常远大于图像元素的宽度，如果直接显示该图像必然会产生一幅模糊的图像。1975年格林(Glenn)开发了一种独特的解决矢量平面问题的方法，该方法第一步是去模糊，第二步再显示矢量断面图像。

另一个方法是首先检测由密度信息表示的物体表面，然后移去表面隐藏线或阴影得到一幅透视图。1977年，霍尔曼(Herman)和刘(Liu)使用这种方法成功地显示了重建图像。1976

年，克里斯芬森(Christenson)和史蒂芬森(Stephenson)及其他人开发了阴影图像显示法，使得阴影透视图的显示变得很容易。图 7-16 和图 7-17 显示了一个胰岛素图像重建的例子。

1. 真实感显示

近年来，计算机图形学的发展极大的促进了图像三维重建技术的发展。图像三维重建技术与计算机图形学的结合使得重建的三维图像极具真实感。真实感显示的关键技术是浓淡层次和光照模型的运用。

浓淡层次和光照特性的显示是对真实世界的一种逼近处理，通常情况下逼真性越强，所采用的光照模型越复杂，计算量也就越大。因此，在实际应用中要兼顾考虑效果与开销。

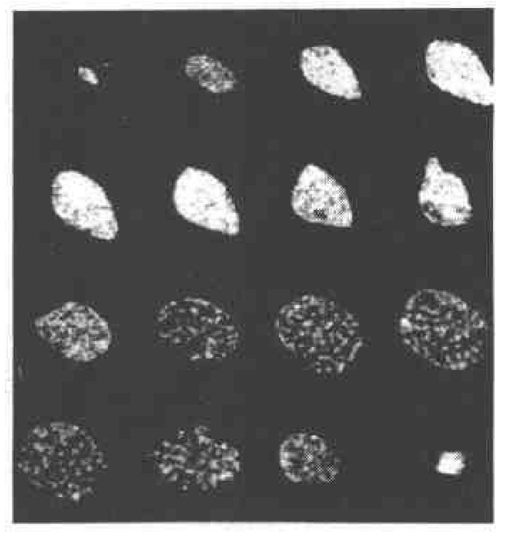


图 7-16 胰岛素切片图

三维重建中的光照模型主要有二个主要成份，重建物体的表面特性与照明特性。表面特性又包括物体的表面反射特性和透明特性。反射特性确定照射到物体表面的光有多少被反射，当物体表面对不同波长的光具有不同的反射系数时，就会出现不同的颜色。透明性确定有多少光线从物体中透射过去，对于透明物体，其颜色由透射光决定。照明特性在浓淡处理中与物体表面特性有同等的重要性。如果照明光源是来自各个方向的均匀光，该种光源称之为漫反射光。如果光源是点光源，物体表面会出现高光效应。除此之外，在照明效果中还会出现光线被遮挡的阴影效应。

一般情况下，光照模型包括局部光照模型和整体光照模型。局部光照模型只考虑光源的漫反射和镜面反射，而整体光照模型要考虑物体间的相互影响，光在物体间的多重吸收，以及反射和透射。较著名的局部光照模型有 Torrance 和 Sparrow 1967 年提出的 Torrance-Sparrow 光照模型，Phong 于 1973 年提出的 Phong 光照模型，Cook 和 Torrance 于 1981 年提出的 Cook-Torrance 光照模型等。目前，实用的整体光照模型有 Whitted 光照模型、Hall 光照模型、双向光线跟踪和分布式光线跟踪等。

2. 简单的光照模型

光线照射到物体表面时，它可以被吸收、反射或透射。被吸收的光能转化为热能，而被反射或透射的光能才能使物体可见并呈现颜色。反射光决定于光的成分、光源的几何性质以及物体表面的方向和表面的性质。反射光分为漫反射光和镜面反射光。漫反射光是光穿过物体表面被吸收后重新发射出来的光，它均匀地分布在各个方向，因此，观察者的观察位置是无关紧要的。镜面反射光由物体的外表面反射所产生。

Lambert 余弦定律总结了点光源照射在完全漫反射体上的光的反射规则，根据这一定律，一个完全的漫反射体上反射出来的光强度同入射光与物体表面法线间夹角的余弦成正比。即：

$$I = I_1 \times K_d \times \cos\theta \qquad 0 < \theta < \frac{\pi}{2} \tag{7-103}$$

式中，I 为反射光的强度，I_1 为从一点光源发出的入射光的光强，K_d 为漫反射系数($0<K_d<1$)，θ 为指向光源方向 L 与表面法向量 $\boldsymbol{n}$ 之间的夹角。如图 7-18 所示，当 $\theta>\frac{\pi}{2}$ 时，光源位于物体后面，式 (7-103) 无意义。漫反射系数 K_d 取决于物体表面材料。反射光强是光波长的函数，在简单光照模型中，通常假定与光波长无关。根据 Lambert 漫反射光照模型绘制的物体表面显

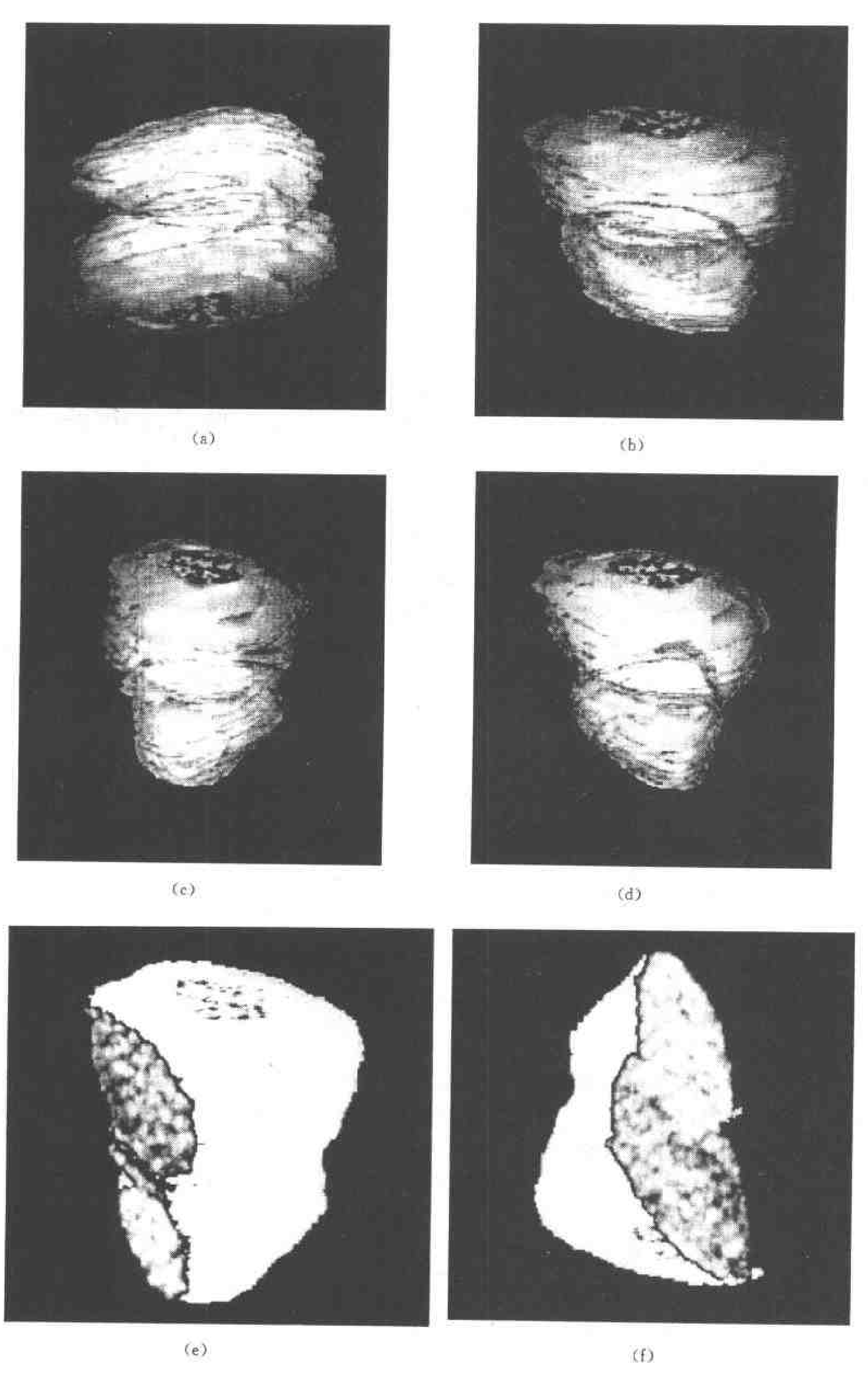


图 7-17 (a)由切片重建的胰岛素图像,(b)、(c)、(d)为不同方向观察的图像、(e)、(f)为胰岛素立体图经剖切后的图像

得比较暗淡，在实际环境中，物体还会受到从周围景物散射光的影响，这是一种亮度均匀的光线，它由光线经过多个面多重反射形成的。通常在应用中把它作为一种常数漫反射项与 Lambert 漫反射分量相加处理。此外，为了反映出物体离光源的远近，还须加入距离修正因子，这样一来，可得到一个简单的光照模型：

$$I = I_a \times K_a + I_1 \times K_d \times \cos\theta / R^2 \tag{7-104}$$

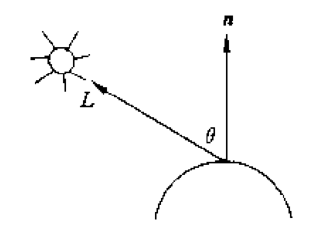


图 7-18 漫反射示意图

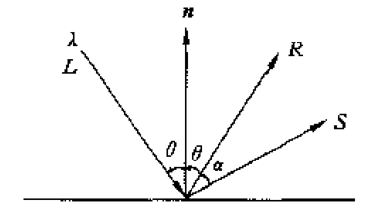


图 7-19 镜面反射示意图

式中 I_a 为入射光的环境光线强度，K_a 为环境的漫反射系数，$0<K_a<1$，R 为物体表面上的一点到光源的距离。

对于平行投影来说，光源在无穷远处，R 无穷大；而对于透射投影，由于视点较靠近物体，$1/R^2$ 就可能出现很大的值，为了使 $1/R^2$ 处于一个较合适的范围而又简化运算，可用 $r+k$ 来替换 R^2，这里 k 为常数，r 是实体表面上的一点 $P(x,y,z)$ 到点光源 (x_0,y_0,z_0) 的距离：

$$r = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2} \tag{7-105}$$

由此，可得到一个改进的光照模型：

$$I = I_a \times K_a + I_1 \times K_d \cos\theta / (r+k) \tag{7-106}$$

如果物体是带有颜色的，可用该光照模型对红、绿、蓝三基色分别计算。

在任何有光泽的表面都会有镜面反射，其光强决定于入射光的角度、入射光的波长及反射表面材料的性质。镜面反射可用菲涅尔公式描述：

$$I = I_a K_a + I_1 (K_d \cos\theta + w(\theta,\lambda)\cos^n\alpha)/(r+k) \tag{7-107}$$

其中，$w(\theta,\lambda)$ 为反射率曲线，它是入射角 θ 和光波长 λ 的函数，n 为幂次，用以模拟反射光的空间分布。α 为视线与反射光线间的夹角。图 7-20 给出了不同的 n 值的 $\cos^n\alpha$ 曲线 $(-\pi/2<\alpha<\pi/2)$。n 越大，表示物体表面越光滑，其分布特性为会聚型的。相反，n 较小表示物体表面粗糙，其分布特性为扩散型的。图 7-21 给出了随入射角变化的几种材料的镜面反射系数。式(7-107) 中函数 $w(\theta,\lambda)$ 较为复杂，在实际使用中，常常根据美学观点或实验数据用一常数 K_s 来代替，从而得到一个简化的光照模型：

$$I = I_a K_a + I_1 (K_d \cos\theta + K_s \cos^n\alpha)/(r+k) \tag{7-108}$$

对于多个点光源，可将它们的效果线性叠加，即：

$$I = I_a K_a + \sum_{j=1}^{m} I_{1,j} (K_d \cos\theta_j + w(\theta,\lambda)\cos^n\alpha_j)/(r_j+k) \tag{7-109}$$

式中，m 为点光源数目，其中：

$$\cos\theta = \boldsymbol{n} \cdot \boldsymbol{L} / (|\boldsymbol{n}| \cdot |\boldsymbol{L}|) = \boldsymbol{e}_n \cdot \boldsymbol{e}_L \tag{7-110}$$

这里：$\boldsymbol{e}_n$，$\boldsymbol{e}_L$ 分别为沿表面法线和光源入射方向的单位矢量。因此，单个光源的光照模型的矢量形式为

$$I = I_1 (K_d (\boldsymbol{e}_n \cdot \boldsymbol{e}_L) + K_s (\boldsymbol{e}_R \cdot \boldsymbol{e}_s)^n + I_a K_a \tag{7-111}$$

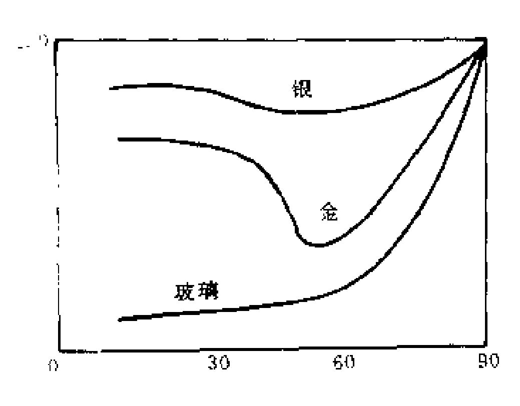



图 7-20 反射率曲线

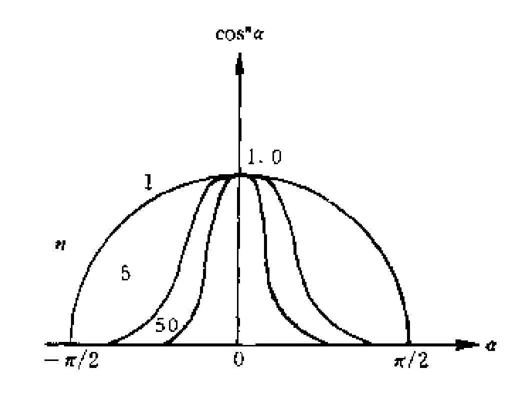


图 7-21 镜面反射光的近似空间分布函数

由于高速处理器对除法运算比较费时，而上式要做开方和除法运算，因此，开销较大。为了克服这一缺点，对上述模型可作如下修正，其条件是把点光源视为在 z 轴无穷远处的透视投影，设实体所有顶点的 z 分量的最小值为 z_{min}，则当观察点位于 z 轴正向时，$z-z_{min}$ 的变化趋势与 $1/(r+k)$ 的变化趋势是一致的。这样，利用相空间中的深度信息来修正光强，可得到实际使用的光照模型：

$$I = K_r(z - z_{min})[K_d(e_n \cdot e_L) + K_s(e_R \cdot e_s)^n] + I_a K_a \tag{7-112}$$

式中 K_r 是常数比例因子，以保证计算出的浓淡值限定在显示灰度范围内。

3. 曲面法向量的计算

重建图像表面法线的方向代表了表面的局部弯曲性，因此，它决定了镜面反射的方向。如果已知重建物体表面的解析表达式，表面法线可直接计算出来。但是，一般情况下，仅知道多边形的近似表示，所以，对每一个多边形小片根据其所在的平面方程的系数决定该小片的法线，并取其外法线方向。通过这样的方法可用多边形的顶点或棱边信息近似求得该顶点的法线。如果每一个多边形的平面方程已知，则多边形顶点的法线取包围该顶点的各多边形的法线的平均值。如图 7-22 所示。

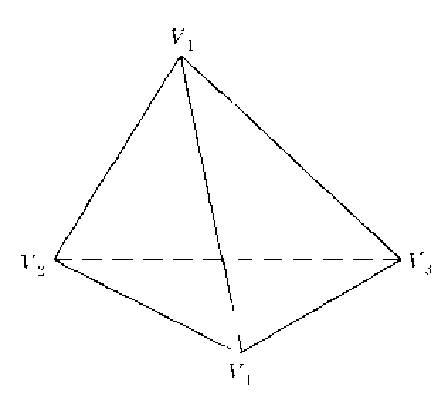


图 7-22 顶点法矢量的近似值

顶点 V_2 的近似法线方向为

$$n_1 = (a_1 + a_2 + a_3)i + (b_1 + b_2 + b_3)j + (c_1 + c_2 + c_3)k \tag{7-113}$$

式中，(a_1, a_2, a_3)，(b_1, b_2, b_3)，(c_1, c_2, c_3)是包围 V_2 的 3 个三角形 ΔV_1、ΔV_2、ΔV_2、ΔV_1、ΔV_3、ΔV_1、ΔV_1、ΔV、ΔV 的平面方程系数。如果各多边形的平面方程未知，顶点处的法线可取交于此顶

点的各棱边的平均值。如图 7-23 所示，顶点 V_1 处的近似法线取为

$$\boldsymbol{n}_1 = \overline{V_1V_2} \times \overline{V_1V_4} + \overline{V_1V_4} \times \overline{V_1V_3} + \overline{V_1V_3} \times \overline{V_1V_2} \tag{7-114}$$

这里，由于只求法线方向，所以，在式(7-123)和式(7-124)中没有将各矢量之和除以包围该点的多边形的个数。

4. 明暗处理算法

在图像重建处理中，光滑表面常用多边形予以近似表示，由于平面上所有点的法向量相同，不同的平面块之间存在着不同的法向量跳跃，从而引起不连续的光亮度跳跃。这样一来，必然会出现相邻的明暗度差别，从而导致 Mach(马赫)效应。为消除亮度的不连续性，1971 年，Gourand 提出了亮度插值明暗处理法，使这种亮度的不连续性有所改善。Gourand 明暗处理法是首先求出各面顶点处的法向量(即求包围该点的各平面的法向量的平均值)，同时计算其浓淡值。多边形面内的浓淡值通过对顶点的浓淡值进行双线性内插求得。如图 7-23 所示的多边形，扫描线与其边界交于 L 和 R，L 处的浓淡值是 A、B 处浓淡值的线性插值。R 处的浓淡值

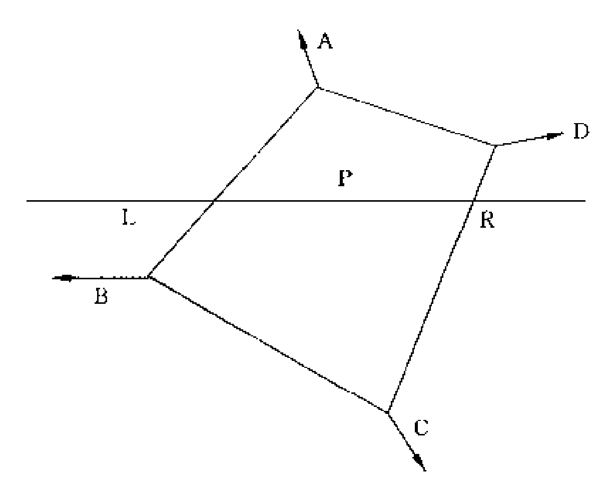


图 7-23　Gourand 法处理示意图

是 C、D 处浓淡值的线性插值。即：

$$I_L = I_A + (1 - \mu)I_B \tag{7-115}$$

$$I_R = I_{DA} - (1 - \upsilon)I_C \tag{7-116}$$

$$I_P = I_L + (1 - \tau)I_R \tag{7-117}$$

其中 I_A、I_B、I_C、I_D、I_L、I_R、I_P 分别为图 7-23 中对应各点的光强度值：μ、υ、τ 分别为下列值：

$$\mu = \frac{BL}{AB} \qquad \upsilon = \frac{CR}{DC} \qquad \tau = \frac{RP}{LR} \tag{7-118}$$

Gourand 法处理方法简单，但它无法模拟高光效果，只实用于简单的漫反射。另一种有效的明暗处理方法是 Phong 法(法向量插值明暗法)。该方法是先计算多边形各顶点处的法向量，再用双线性插值的方法求得每个像素处的法向量，最后对每个像素所得到的法向量用 Phong 光照模型求出明暗值。即：

$$I = K_a I_a + K_d I_e \cos\theta + K_s I_l \cos^n \alpha \tag{7-119}$$

式中：最后一项用于模拟镜面反射光，以便能再现高光。由于插值是基于描述物体表面朝向的法向量，所以，Phong 方法可较好地在局部范围内模拟表面的弯曲性，可得到较好的曲面绘制结果，镜面反射模拟高光的效果显得更加真实。具体计算中，一般不用 $\cos\alpha$，而用点积 $\boldsymbol{N} \cdot \boldsymbol{H}$。

其中 N 是物体表面法向量，H 是 L 和 E 的平均向量再单位化，即：$H=(L+E)/|L+E|$。Phong 模型所涉及的几何向量如图 7-24 所示。

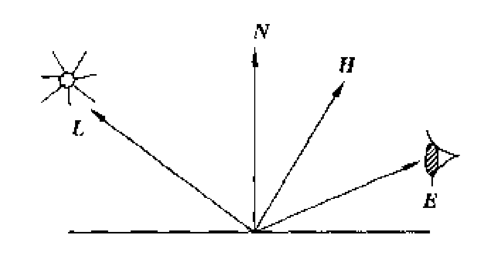


图 7-24 Phong 模型所涉及的几何向量

当物体是透明的时候，不但会有反射光，而且还有透射光。所以，会透过物体看到后面的东西。一般来说，光通过不同的介质表面时，会发生折射，也就是会改变传播方向。为了模拟折射，需要较大的计算量。如果忽略折射，会得到一种最简单的生成透明物体的方法。因为忽略了折射效应，所以光通过物体表面时不发生方向的改变。如图 7-25 所示，图中如果考虑折射，则 A 点可见，如果不考虑折射，则 B 点可见。

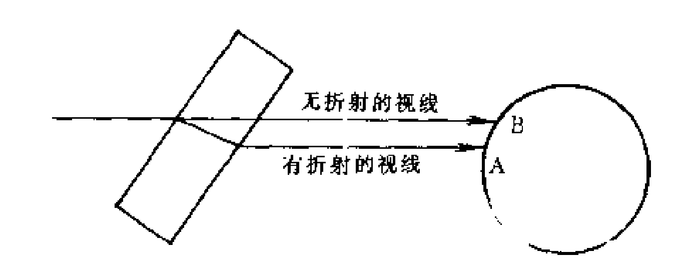


图 7-25 有折射与无折射的视线差别

一般的隐面消除算法均实用于模拟不考虑折射的透明的情况。例如，当利用扫描线算法生成物体图形时，假设视线交于一个透明的物体表面后再交于另一物体表面，在两个交点的明暗度分别是 I_1 和 I_2，则可以把综合光强表示为两个明暗度的加权和，即：

$$I = KI_1 + (1 - K)I_2 \tag{7-120}$$

式中 K 是第一个物体的透明度，$0 \leqslant K \leqslant 1$，在极端情况下，$K=0$。第一个物体表面完全透明，因此，对后面的物体完全没有影响。而当 $K=1$ 时，则物体是不透明的，则后面的物体被遮挡，对当前像素的明暗度不产生影响。

上面介绍的是在三维重建中为增加真实感而采用的光照效果的模拟，如果只考虑光照效果，在有些情况下显得还是不够逼真，因为自然界的物体大都有各种特有的纹理。为使重建物体更加逼真，纹理映射处理也是十分必要的。该技术包括纹理映射算法、纹理反走样处理等相关技术，经纹理映射处理后，重建物体的真实感将大大加强。关于纹理映射处理的原理及算法在计算机图形学中都有详细的论述，这里就不再赘述。

图像三维重建技术经科技工作者的辛勤工作，近几年有了突飞猛进的发展，在许多领域都有重要应用。特别是与计算机图形学相结合，产生了许多新的算法，重建图像的真实感有了质的提高。这正体现了当前科技发展的总特点——多个学科、多种技术互相融合、渗透、移植、借鉴、综合的优势，由此而产生的科技成果必将造福于全人类。

7.7.4 图像重建的应用

图像重建技术在许多科学领域得到广泛应用，其中最为显著的是医学方面的应用。根据原始数据获取方法及重建原理的不同可分为如下几种：

放射断层重建成像(Emission Computed Tomography，ECT)：这种方法是在物体中注入放射性物质，然后，从物体外部检测通过物体后放射出来的能量，由于物体内部不同的物质或人体不同的组织对放射能量有不同的吸收或衰减，由此可获取不同的数据，用这些数据就可以重建出所需要的图像，从而达到检测物体内部的分布情况或人体内部的病变。

透射断层重建成像(Transmission Computed Tomography，TCT)：这种方法是射线源的射线穿过物体或人体组织，然后由接受器接受经过物体或人体组织吸收后的剩余射线，由于不同的物质或组织对射线的吸收不同，所以，剩余的射线反映了物体内部的状况和人体内部的不同组织，通过这些数据再重建图像，从而发现物体内部缺欠或人体内部病变。

反射断层重建成像(Reflection Computed Tomography，RCT)：该方法常用于雷达系统。雷达图像通常是由物体反射的回波产生的。

核磁共振重建成像(Magnetic Resonance Imaging，MRI)：该方法是由于具有奇数个质子或中子的原子核包含有一定磁动量或旋量的质于，如果把它们放在磁场中，它们就会像陀螺在重力场中一样运动，在一般情况下，质子在磁场中任意排列，当有适当强度和频率的共振场作用于物体时，质子吸收能量并转向与磁场相交的方向。如果此时把共振磁场去掉，则质子吸收的能量释放并被检测器收集，根据检测的信号就可以确定质子的密度。通过控制共振磁场的强度，可检测到一条直线上的信号，从而通过该信号数据可重建出物体图像。

以上是图像重建的几种方法，这些方法有的是人们所熟悉的在医学中广泛应用的方法。下面就医学应用介绍一些基本图像重建的原理。

在1968—1972年英国的EMI公司的Hounsfield研制了头部CT，发明了计算机断层摄影术。这是图像处理技术对医学领域的杰出贡献。它第一次用X射线来作为信息收集源，把X射线学应用于临床诊断。在该技术中，用横向运动技术，开发了一种计算机断层摄影术——简称CTAT或CAT的技术。近年来，计算机断层摄影术更广泛的应用于核医学来描述正电子发射的同位素图像。我们把这一技术称为放射计算机摄影术(ECT)。

计算机断层摄影术是随着著名的针对脑部的X射线扫描系统的引入而开始发展起来的一种技术。这个系统由EMI公司的Hounsfield于1971年开发出来的。接下来，Terpogossian和Phelps把计算机断层摄影术的原理应用到核医学图像处理中。在核医学图像处理中，他们对断层摄影系统在用正电子发射使同位素衰减方面的应用作了进一步的发展。1975年，Budinger又把计算机断层摄影术的原理应用到腹部图像重建中。

图7-26显示了X射线系统的基本装置。在传统的辐射学中，从三维物体中得到的信息可用二维的形式叠加在胶片上。通常，诊断中收集到的信息需要增强处理，以提高对比度。计算机断层摄影术是指图像数据是从物体横断面上直接截取。在这一横断面上，一个虚构的图像矩阵是由事前规定了尺寸的方形元素组成的。在X射线脑部成像系统中，元素的大小为1～3mm^2。通常，为了使图像能包含目标物，图像矩阵应足够大。在脑部的检测中，典型的大小是148个元素，约有25cm。

为了收集数据并重建图像，该装置把X射线源构成的检测器排成一行，装置的X射线具有一个像素宽度并高度平行。扫描动作的初始，横切运动是一个线性的运动，其中每148行(每

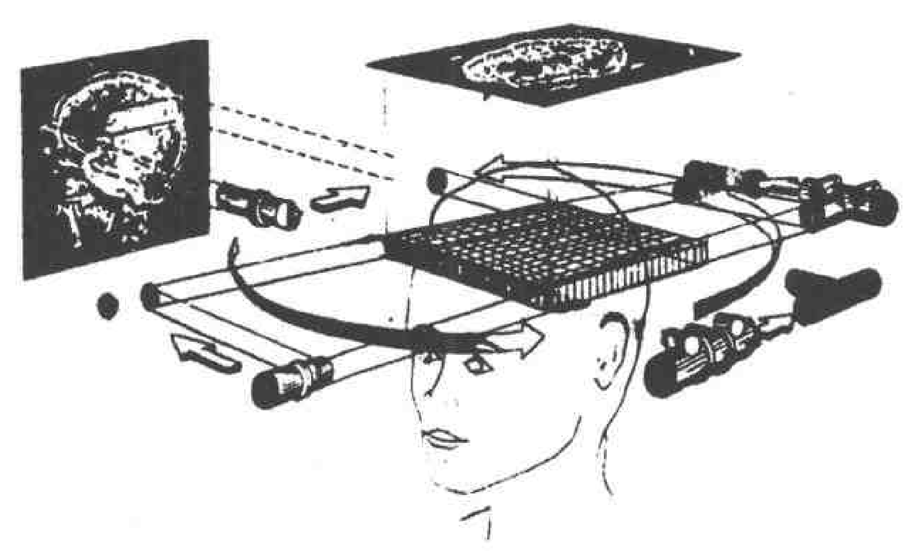


图 7-26　X 射线计算机断层系统原理图

行包含 148 个以上的元素)分成一块。这样,在这一横切过程中 148(或更多)个离散点数据被输入计算机。一个横切过程完成后,整个 X 射线源检测器的几何位置就要旋转到一个新的位置(假设每次增加值定为 1°),于是,下一个线性的横切扫描运动又重新开始。如果 180°角的投影都用到了,则结果数据为 148 的 180 倍,即 26640 个离散的投影图数据就被输入到计算机内。在扫描运动进行时,数据以数字的形式存入计算机。X 射线计算机断层摄影术的一个重要功能就是最终图像的信息密度是可控制的。人体组织对 EMI 扫描器 X 射线的吸收系数是不同的,变化范围大约在百分之几,这取决与人体的不同组织如:脂肪、肌肉或其他组织。但是,辐射的统计量必须满足要求,即每一个数据点必须包含 4000～50000 计数,这一数量从被扫描的物体得到并由 X 射检测器收集。

与 gamma 照相机或核医学扫描仪产生的信息量相比,放射性图片有非常高的信息密度。计算机断层摄影图则在这二者之间。在扫描的同时,数字数据被输入计算机以形成最终的图像。在该技术中,横切运动和用于计算机断层摄影系统的 X 射线检测器是用一个胶卷所替代。显然,胶片必须放在一个可以通过物体来接收 X 射线的角度上。关键的一点是胶片只能记录它"看见"的东西。因此,图像附近的东西也叠加上去了,结果使图像质量较差,图像模糊。图 7-27 示出了透射图像的重建处理过程。这里,设计了一个有不同的吸收参数的模拟剖视物体图像,如图 7-27(a)所示。同心环的高度分别选为底座的 20%、2%、50%。利用计算机程序来产生投影,该投影是在 x 光传输系统中通过实验产生。这样一组投影产生的角度从 0°～180°。这些图像首先叠加起来模拟该物体。如图 7-27(b)所示立体图。可以看到,50%高度的圆环部分和底座是可见的。而 2%和 20%高度的几层圆环几乎辨别不出来。X 射线的简单叠加产生一个几乎无法辨别的图像,其剖切图也不正确。



该模型图像的正确重建结果应是如图 7-27(c)所示。图 7-27(c)应用了加权因子与去模糊处理。我们发现该图轮廓清晰,不但 20%高度的那部分清晰可辨,而且 2%的那部分也十分清晰。

上述技术可用于利用直角几何与扇形几何的透射系统实现,也可以利用直角发射和环形检测器系统。

以上只是就图像重建技术在医学中的一种应用，类似的应用还有很多，如正电子发射同位素计算机断层摄影术的应用等。它们采用的重建原理大体上是相同的，只不过在重建数据的产生和采集原理及装置不同罢了。

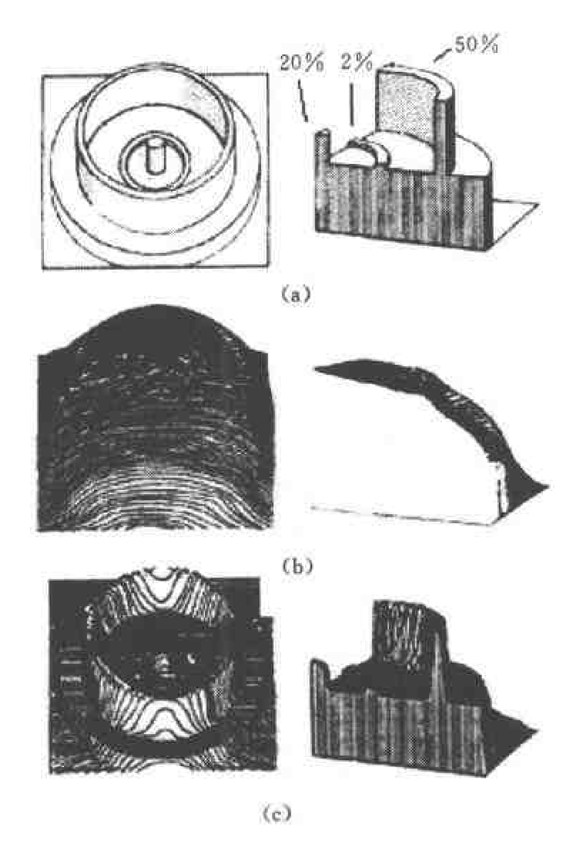


图 7-27 透射图像处理

(a)原模拟物体，(b)简单迭加处理结果，(c)采用加权因子和去模糊处理的结果

思考题

1. 图像重建中数据的获取模型有哪几种？
2. 试述傅里叶重建法的基本原理。
3. 试述卷积法图像重建的基本原理。
4. 试述代数法重建的基本原理。
5. 重建优化问题有什么意义？
6. 图像重建中滤波器的作用如何？试用传统的滤波器处理之，并观察其效果。
7. 试述重建图像显示中 γ 校正的意义。
8. 光照模型有几种？试述其原理及作用。

第8章 图像分析

对图像进行增强、恢复、编码等处理时，输入是图像，所要求的输出是一幅近似于输入的图像，这是此类处理的一个特点。图像处理的另一个主要分支是图像分析或景物分析，这类处理的输入仍然是图像，但是所要求的输出是已知图像或景物的描述。这类处理基本上用于自身图像分析和模式识别一类的领域。例如，染色体的分类、排列，血球的分类、计数，航空照片的地貌分类以及机器人的识别系统等等。描述一般是针对图像或景物中的特定区域或目标。为了描述，首先要进行分割，有些分割运算可直接用于整个图像，而有些分割算法只适用于已被局部分割的图像。例如，分割染色体的处理，可先用设置门限的方法把染色体和背景分割开来，然后可采用尺寸大小、形状等准则进一步将其分割成单个染色体。

值得注意的一点是，没有惟一的、标准的分割方法，因此，也就没有规定成功分割的准则。本章只讨论一些最基本的分割、描述方法。

8.1 分割

分割的目的是把图像空间分成一些有意义的区域。例如一幅航空照片，可以分割成工业区、住宅区、湖泊、森林等等。可以以逐个像素为基础去研究图像分割，也可以利用在规定领域中的某些图像信息去分割。分割的依据可建立在相似性和非连续性两个基本概念之上。

8.1.1 灰度阈值法分割

最常用的图像分割方法是把图像灰度分成不同的等级，然后用设置灰度门限的方法确定有意义的区域或欲分割的物体之边界。

假定一幅图像具有图8-1所示的直方图。由直方图可以知道图像 $f(x,y)$ 的大部分像素灰度值较低，其余像素较均匀地分布在其他灰度级上。由此可以推断这幅图像是由有灰度级的物体叠加在一个暗背景上形成的。可以设一个阈值 T，把直方图分成两个部分，如图所示。T 的选择要本着如下原则：B_1 应尽可能包含与背景相关连的灰度级，而 B_2 则应包含物体的所有灰度级。当扫描这幅图像时，从 B_1 到 B_2 之间的灰度变化就指示出有边界存在。当然，为了找出水平方向和垂直方向上的边界，要进行两次扫描。也就是说，首先确定一个门限 T，然后执行下列步骤。

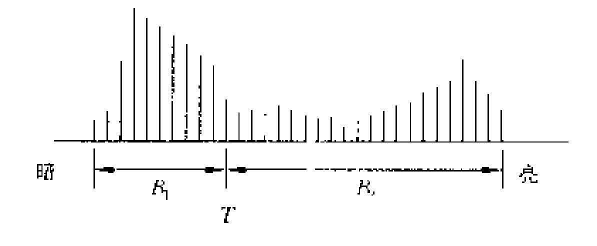


图8-1 图像 $f(x,y)$ 的直方图

第一,对 $f(x,y)$ 的每一行进行检测,产生的图像的灰度将遵循如下规则:

$$f_1(x,y)=\begin{cases}L_E & f(x,y)\text{ 和 }f(x,y-1)\text{ 处在不同的灰度级上}\\ L_B & \text{其他}\end{cases} \tag{8-1}$$

式中 L_E 是指定的边缘灰度级,L_B 是背景灰度级。

第二,对 $f(x,y)$ 的每一列进行检测,产生的图像的灰度将遵循下述规则:

$$f_2(x,y)=\begin{cases}L_E & f(x,y)\text{ 和 }f(x-1,y)\text{ 灰度处在不同的灰度级上}\\ L_B & \text{其他}\end{cases} \tag{8-2}$$

为了得到边缘图像,可采用下述关系:

$$f(x,y)=\begin{cases}L_E & f_1(x,y)\text{ 或 }f_2(x,y)\text{ 中的任何一个等于 }L_E\\ L_B & \text{其他}\end{cases} \tag{8-3}$$

上述方法是以某像素到下一个像素间灰度的变化为基础的。这种方法也可以推广到多灰度级阈值方法中。由于确定了更多的灰度级阈值,可以提高边缘抽取技术的能力,其关键问题是如何选择阈值。

一种方法是把图像变成二值图像。例如,图像 $f(x,y)$ 的灰度级范围是 (z_l,z_h),设 T 是 z_l 和 z_h 之间的一个数,那么 $f_1(x,y)$ 可由式(8-4)表示:

$$f_1(x,y)=\begin{cases}1 & f(x,y)\geqslant T\\ 0 & f(x,y)<T\end{cases} \tag{8-4}$$

另一方法是把规定的灰度级范围变换为 1,而把范围以外的灰度级变换为 0,例如

$$f_u(x,y)=\begin{cases}1 & f(x,y)\leqslant u\\ 0 & f(x,y)>u\end{cases} \tag{8-5}$$

$$f_{u,v}(x,y)=\begin{cases}1 & u\leqslant f(x,y)\leqslant v\\ 0 & \text{其他}\end{cases} \tag{8-6}$$

另外,还有一种所谓半阈值法,这种方法是将灰度级低于某一阈值的像素灰度变换为 0,而其余的灰度级不变,仍保留原来的灰度值。总之,设置灰度级阈值的方法不仅可以提取物体,也可以提取目标物的轮廓。这些方法都是以图像直方图为基础去设置阈值的。

那么,在分割中如何设置最佳阈值呢?假设一幅图像是由背景和物体组成。其中,物体像素的灰度级具有正态概率密度 $p(z)$,其均值为 μ,方差为 σ^2;而背景像素的灰度级也具有正态概率密度 $q(z)$,其均值为 v,方差为 τ^2。物体占图像总面积的比为 θ,背景占总面积的比为 $1-\theta$,所以这幅图像总的灰度级概率密度为

$$\theta p(z)+(1-\theta)q(z) \tag{8-7}$$

假设对图像设置一阈值 t,并且把小于 t 的全部点称为目标物体点,而把大于等于 t 的所有点称为背景点。那么,把背景错归为物体点的概率为 $Q_1(t)$,把物体点错归为背景点的概率为 $Q_2(t)$,则有

$$Q_1(t)=\int_{-\infty}^{t} q(z)\mathrm{d}z \tag{8-8}$$

$$Q_2(t)=\int_{t}^{\infty} p(z)\mathrm{d}z=1-p(t) \tag{8-9}$$

总的错分概率为

$$\theta Q_2(t)+(1-\theta)Q_1(t)=\theta[1-p(t)]+(1-\theta)Q_1(t) \tag{8-10}$$

要求得式(8-10)的最小阈值,可将上式对 t 微分,并令其结果为 0,则得到

$$(1-\theta)q(t)=\theta p(t) \tag{8-11}$$

因为

$$p(t)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right] \tag{8-12}$$

$$q(t)=\frac{1}{\sqrt{2\pi}\tau}\exp\left[-\frac{(t-\upsilon)^2}{2\tau^2}\right] \tag{8-13}$$

代入式(8-11)，并取对数

$$\ln\sigma+\ln(1-\theta)-\frac{(t-\upsilon)^2}{2\tau^2}=\ln\tau+\ln\theta-\frac{(t-\mu)^2}{2\sigma^2} \tag{8-14}$$

或者

$$\tau^2(t-\mu)^2-\sigma^2(t-\upsilon)^2=2\sigma^2\tau^2\ln\frac{\tau\theta}{\sigma(1-\theta)} \tag{8-15}$$

由这个二次方程可以求解出 t 值。如果 $\theta=1/2$，$\tau=\sigma$，那么

$$t=\frac{(\mu+\upsilon)}{2} \tag{8-16}$$

这一结果可以证明如下：

如果知道图像点的 θ 部分是物体点，那么可以设一个分位数 θ，即设置一个阈值 t，使像点的一部分 θ 具有小于 t 的灰度级，因此，有

$$\int_{-\infty}^{t}[\theta p(z)+(1-\theta)q(z)]\mathrm{d}z=\theta \tag{8-17}$$

在 $\theta=1/2$ 和 $\tau=\sigma$ 情况下，密度函数：

$$\mathrm{d}z=\frac{1}{\sqrt{2\pi}\sigma}\left\{\exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right]+\exp\left[-\frac{(z-\upsilon)^2}{2\sigma^2}\right]\right\}$$

在点 $z=\dfrac{(\mu+\upsilon)}{2}$ 周围是对称的，即对任何 ω 有

$$\mathrm{d}\left(\frac{\mu+\upsilon}{2}+\omega\right)=\mathrm{d}\left(\frac{\mu+\upsilon}{2}-\omega\right) \tag{8-18}$$

因此，$\dfrac{\mu+\upsilon}{2}$ 是 $\mathrm{d}z$ 的中位数。将 $\theta p(z)+(1-\theta)q(z)$ 式对 z 微分，并令其结果为零可找出该式的极大值和极小值，即

$$\theta\tau^3\exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right](z-\mu)+(1-\theta)\sigma^3\exp\left[-\frac{(z-\upsilon)^2}{2\tau^2}\right](z-\upsilon)=0 \tag{8-19}$$

在特殊情况下，式(8-19)可简化为式(8-20)的形式：

$$\exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right](z-\mu)+\exp\left[-\frac{(z-\upsilon)^2}{2\sigma^2}\right](z-\upsilon)=0 \tag{8-20}$$

此式在 $z=\dfrac{(\mu+\upsilon)}{2}$ 处有一个根。这个根是对应的极大值还是极小值需验证 $\mathrm{d}z$ 的导数。在忽略共同的正常数因子的情况下，它的一次导数为

$$\upsilon(z)=-(z-\mu)\exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right]-(z-\upsilon)\exp\left[-\frac{(z-\upsilon)^2}{2\sigma^2}\right] \tag{8-21}$$

对于 $z<\mu$，则式(8-21)两项均为正，$z>\upsilon$，这两项都为负。因此，当 $z<\mu$ 时，$\mathrm{d}z$ 严格递增，而 $z>\upsilon$ 时，$\mathrm{d}z$ 严格递减。另外，

$$\mathrm{d}''(z)=\left[\frac{(z-\mu)^2}{\sigma^2}-1\right]\exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right]+\left[\frac{(z-\upsilon)^2}{\sigma^2}-1\right]\exp\left[-\frac{(z-\upsilon)^2}{2\sigma^2}\right] \tag{8-22}$$

同样，略去共同的正常数因子，当 $z=\frac{(\mu+\upsilon)}{2}$时，此式取值为

$$2\left[\frac{(\upsilon-\mu)^2}{4\sigma^2}-1\right]\exp\left[-\frac{(\upsilon-\mu)^2}{8\sigma^2}\right] \tag{8-23}$$

这样，如果$(\upsilon-\mu)^2>4\sigma^2$，则 $d''\left(\frac{\mu+\upsilon}{2}\right)$为正。那么 dz 在 $z=\frac{(\mu+\upsilon)}{2}$处有极小值；且由于 $z<\mu$ 时 dz 递增，$z>\upsilon$ 时 dz 递减，则在 μ 和$\frac{\mu+\upsilon}{2}$之间以及在$\frac{\mu+\upsilon}{2}$和 υ 之间它还必须有极大值。若 $\upsilon-\mu>2\sigma$，阈值$\frac{\mu+\upsilon}{2}$对应于 dz 的两峰间的一个极小值。以上便是在 $\theta=\frac{1}{2}$，$\tau=\sigma$ 的情况下，对 $t=\frac{\mu+\upsilon}{2}$的简单分析。当然，即使概率密度函数 $p(z)$和 $q(z)$不是正态分布，方程$(1-\theta)q(t)=\theta p(t)$仍可用来确定最小误差阈值 t、只要 p 和 q 是已知函数，就可以用数值解法来对 t 求解。

对于复杂图像，在许多情况下对整幅图像用单一阈值不能给出良好的分割结果。例如，图像是在光亮背景上的暗物体，但由于照射光的不均匀，虽然物体与背景始终有反差，但在图像的某一部分物体和背景两者都比另一部分亮。因此，在图像的一部分能把物体和背景精确地分开的阈值，对另一部分来说，可能会把太多的背景也当作物体分割下来了。克服这一缺点有如下一些方法：如果已知在图像上的位置函数描述不均匀照射，就可以设法利用灰度级校正技术进行校正，然后采用单一阈值来分割；另一种方法是把图像分成小块，并对每一块设置局部阈值。但是，如果某块图像只含物体或只含背景，那么对这块图像就找不到阈值。这时，可以由附近的像块求得的局部阈值用内插法给此像块指定一个阈值。

在确定阈值时，如果阈值定得过高，偶然出现的物体点就会被认作背景。如果阈值定得过低，则会发生相反的情况。克服的方法是使用两个阈值。例如，$t_1<t_2$，把灰度值超过 t_2 的像素分类为核心物体点，而灰度值超过 t_1 的像素仅当它们紧靠核心物体点时才算作物体点。t_2 的选择要使每个物体有一些像素灰度级高于 t_2，而背景不含有这样的像素。同时，应选择 t_1 使每个物体像素点具有高于 t_1 的灰度级。如果只使用 t_2，则物体总是分割得不完整；如果只使用 t_1，则会有许多背景像素被错分为物体像素。但是，如果同时使用 t_1 和 t_2 就能把背景与物体很好地分割开来。当然，如果物体与背景的对比是鲜明的，就不必使用这种方法。

此外，如果存在一个阈值 t_2，使得每个物体的像素灰度级高于 t_2，而背景不包含这种像素，可对图像设置阈值 t_2，然后检查高于阈值像素的区域，目的是寻找一个局部阈值，以便在每个类似区域中把物体和背景分开。如果这些物体相当小，并且不太靠近在一起时，这种方法比较适用。所使用的区域应足够大，以保证它们既包含物体像素，也包含背景像素，这样就可以使区域的直方图是双峰的。

有时需要寻找一幅图像的局部最大点，即提取比附近像素有较高的某种局部性质值的像素。一般来讲，也要求这些点具有高于一个低阈值 t_1 的值，一旦超过 t_1，不管它的绝对值大小如何，一切相对的最大值都被采纳。因此，可把寻找局部最大值看作为局部设置阈值的极端情况。在对图像进行匹配运算或检测界线时可采用这种方法。

8.1.2 样板匹配

在数字图像处理中，样板是为了检测某些不变区域特性而设计的阵列。样板可根据检测目的的不同而分为点样板、线样板、梯度样板、正交样板等等。

点样板的例子如图 8-2 所示。下面用一幅具有恒定强度背景的图像来讨论。这幅图像包含

了一些强度与背景不同且互相隔开的小块(点),假定小块之间的距离大于$[(\Delta x)^2+(\Delta y)^2]^{1/2}$,这里$\Delta x$、$\Delta y$分别是在$x$和$y$方向的取样距离,用点样板的检测步骤如下:

−1	−1	−1
−1	8	−1
−1	−1	−1

图 8-2 点样板

样板中心(标号为 8)沿着图像从一个像素移到另一个像素,在每一个位置上,把处在样板内的图像的每一点的值乘以样板的相应方格中指示的数字,然后把结果相加。如果在样板区域内所有图像的像素有同样的值,则其和为零。另一方面,如果样板中心位于一个小块的点上,则其和不为零;如果小块在偏离样板中心的位置上,其和也不为零,但其响应幅度比起这个小块位于样板中心的情况时要小一些,这时,可以采用阈值法清除这类较弱的响应;如果其幅度值超过阈值,就意味着小块被检测出来了;如果低于阈值则被忽略掉。

例如,设$w_1,w_2,\cdots,w_9$代表 3×3 模板的权,并使$x_1,x_2,\cdots,x_9$为模板内各像素的灰度值。从上述方法来看,应求出两个矢量的积,即:

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X} = w_1x_1 + w_2x_2 + \cdots + w_9x_9 = \sum_{n=1}^{9} w_n x_n \tag{8-24}$$

式中

$$\boldsymbol{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_9 \end{bmatrix} \tag{8-25}$$

$$\boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} \tag{8-26}$$

设置一阈值T,如果

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X} > T \tag{8-27}$$

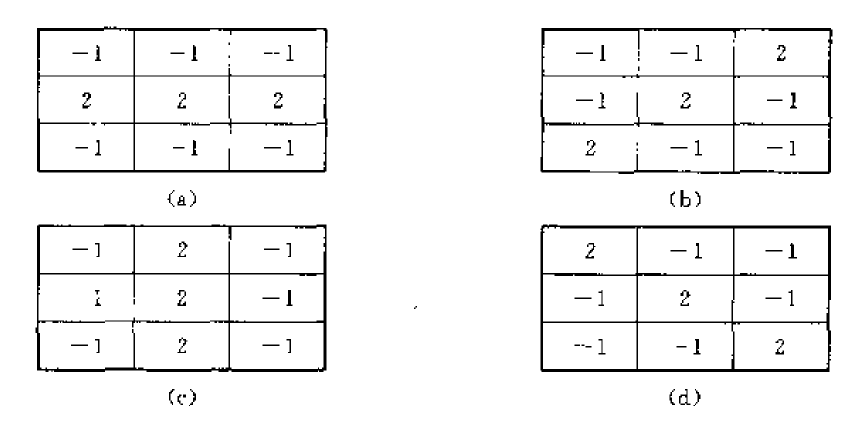
我们认为小块已检测出来了。这个步骤可很容易地推广到$n\times n$大小的样板,不过此时要处理n^2维矢量。

线检测样板如图 8-3 所示。其中,样板(a)沿一幅图像移动,它将对水平取向的线(一个像素宽度)有最强的响应。对于恒定背景,当线通过样板中间一行时出现最大响应;样板(b)对 45°方向的那些线具有最好响应;样板(c)对垂直线有最大响应;样板(d)则对−45°方向的那些线有最好的响应。

设$\boldsymbol{W}_1,\boldsymbol{W}_2,\boldsymbol{W}_3,\boldsymbol{W}_4$是图 8-3 中 4 个样板的权值组成的 9 维矢量。与点样板的操作步骤一样,在图像中的任一点上,线样板的各个响应为$\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X}$,这里$i=1,2,3,4$。此处$\boldsymbol{X}$是样板面积内 9 个像素形成的矢量。给定一个特定的$\boldsymbol{X}$,希望能确定在讨论问题的区域与 4 个线样板中的哪一个有最相近的匹配。如果第i个样板的响应最大,则可以断定$\boldsymbol{X}$和第i个样板最相近。换言之,如果对所有的j值,除$j=i$外,有

$$W_i^{\mathrm{T}}\boldsymbol{X} > W_j^{\mathrm{T}}\boldsymbol{X} \tag{8-28}$$

就可以说 X 和第 i 个样板最相近。如果 $W_i^T X > W_j^T X$，$j=2,3,4$，可以断定 X 代表的区域有水平线的性质。



-1	-1	-1
2	2	2
-1	-1	-1

(a)

-1	-1	2
-1	2	-1
2	-1	-1

(b)

-1	2	-1
1	2	-1
-1	2	-1

(c)

2	-1	-1
-1	2	-1
-1	-1	2

(d)

图 8-3　线样板

对于边缘检测来说也同样遵循上述原理。通常采用的方法是执行某种形式的二维导数。类似于离散梯度计算，考虑 3×3 大小的模板，如图 8-4 所示。

a	b	c
d	e	f
g	h	i

图 8-4　3×3 样板

考虑 3×3 的图像区域，G_x 及 G_y 分别用下式表示：

$$G_x = (g + 2h + i) - (a + 2d + c) \tag{8-29}$$

$$G_y = (c + 2f + i) - (a + 2d + g) \tag{8-30}$$

在 e 点的梯度为

$$G = [G_x^2 + G_y^2]^{\frac{1}{2}} \tag{8-31}$$

采用绝对值的一种定义为

$$G = |G_x| + |G_y| \tag{8-32}$$

梯度模板如图 8-5 所示。

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

图 8-5　梯度样板

把图 8-5 的区域与式(8-29)比较可以看出，G_x 为第一行和第三行的差，其中最靠近 e 的元素(b 和 h)的加权等于角偶上权值的两倍。因此，G_x 代表在 x 方向上导数的估值。式(8-31)和式(8-32)可用图 8-5 中的两个样板来实现。

边缘检测也可以表示成矢量，其形式与线样板检测相同。如果 X 代表所讨论的图像区域，则有

$$G_x = W_1^T X \tag{8-33}$$

$$G_y = W_2^T X \tag{8-34}$$

这里 W_1、W_2 是图 8-5 中的两个样板矢量。W_1^T、W_2^T 分别代表它们的转置。这样，梯度公式(8-31)和式(8-32)变为式(8-35)和式(8-36)的形式：

$$G = [(W_1^T X)^2 + (W_2^T X)^2]^{\frac{1}{2}} \tag{8-35}$$

$$G = |W_1^T X| + |W_2^T X| \tag{8-36}$$

检测点、线和边缘的矢量公式可应用于 1977 年费雷和陈提出的一种检测技术。他们提出的检测方法是这样实现的：假定有两个只有三个元素的样板，此时，则有两个矢量 W_1 和 W_2，它们都是三维的；又假定 W_1 和 W_2 都是正交的和归一化的，因此，它们都有单位幅值。$W_1^T X$ 和 $W_2^T X$ 项分别等于在相应矢量 W_1 和 W_2 上 X 的投影。对于 W_1 来说，

$$W_1^T X = |W_1||X|\cos\theta \tag{8-37}$$

这里 θ 是两个矢量间的夹角。因为 $|W_1| = 1$，因此有

$$|X|\cos\theta = W_1^T X \tag{8-38}$$

这就是 X 在 W_1 上的投影，这种情况如图 8-6 所示。对 W_2 来说亦然。

现在假定有 3 个正交的单位矢量 W_1、W_2、W_3 分别与 3 个三点样板相对应，那么乘积 $W_1^T X$、$W_2^T X$、$W_3^T X$ 代表 X 在 3 个矢量 W_1、W_2、W_3 上的投影。其几何关系如图 8-7 所示。

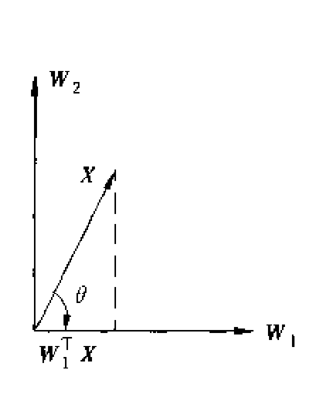


图 8-6　X 向单位矢量 W_1 的投影

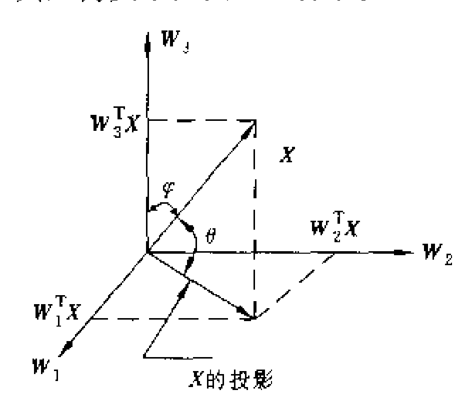


图 8-7　X 向 W_1、W_2、W_3 确定的子空间的投影

假定样板 1 和 2 是检测线的，而 3 是检测点的，X 代表的这个区域是更像一条线呢还是更像一个点呢？为了回答这一问题，把 X 投影到 W_1、W_2、W_3 的子空间上，X 和子空间的夹角可以说明 X 更接近于线还是更接近于点。这可以从图 8-7 的几何关系上看出来。X 在由 W_1 和 W_2 所确定的平面上投影的幅度可由式(8-35)表示，而 X 的幅度由下式表示：

$$|X| = [(W_1^T X)^2 + (W_2^T X)^2 + (W_3^T X)^2]^{\frac{1}{2}} \tag{8-39}$$

X 和其投影间的夹角为

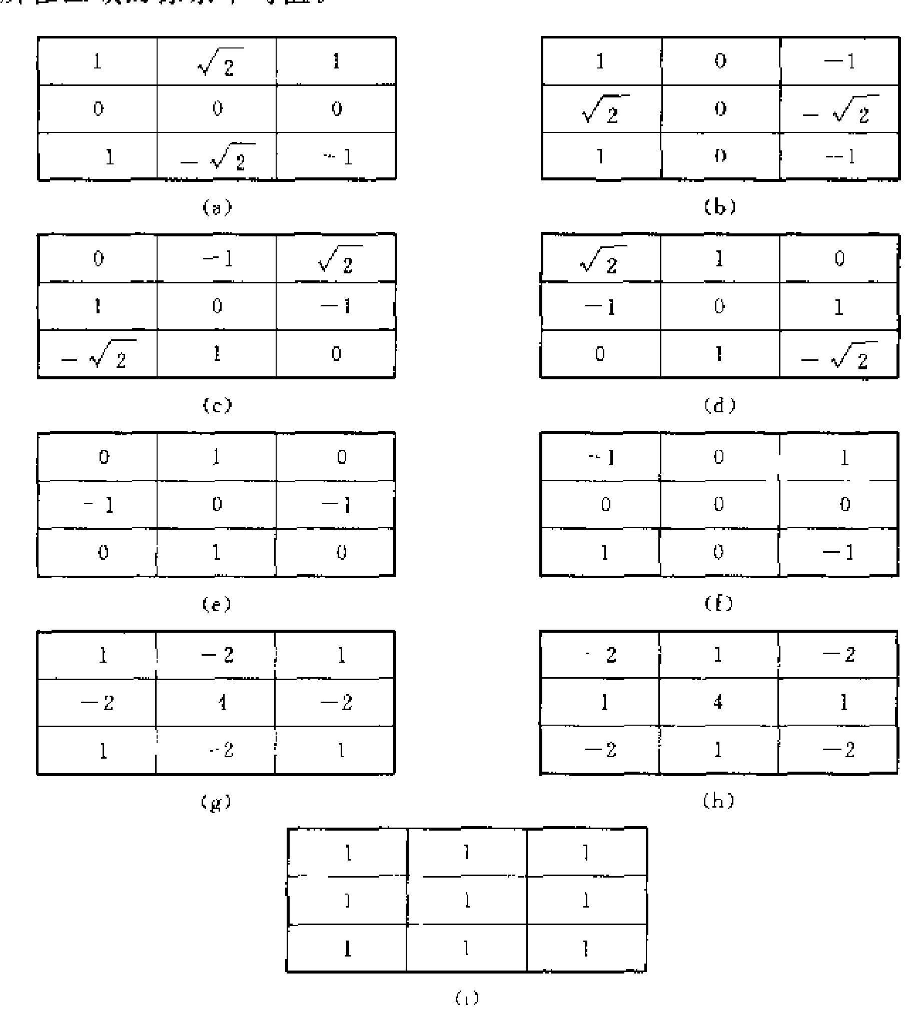
$$\begin{aligned}\theta &= \arccos\left\{\frac{[(W_1^T X)^2 + (W_2^T X)^2]^{\frac{1}{2}}}{[(W_1^T X)^2 + (W_2^T X)^2 + (W_3^T X)^2]^{\frac{1}{2}}}\right\} \\ &= \arccos\left\{\frac{\left[\sum_{i=1}^{2}(W_i^T X)^2\right]^{\frac{1}{2}}}{\left[\sum_{i=1}^{3}(W_i^T X)^2\right]^{\frac{1}{2}}}\right\} = \arccos\left\{\frac{1}{|X|}\left[\sum_{i=1}^{2}(W_i^T X)^2\right]^{\frac{1}{2}}\right\}\end{aligned} \tag{8-40}$$

同理，向 W_3 子空间上投影的夹角可由下式表示：

$$\varphi = \arccos\left\{ \frac{1}{|\boldsymbol{X}|}\left[\sum_{i=3}^{3} (\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X})^2 \right]^{\frac{1}{2}} \right\} = \arccos\left\{ \frac{1}{|\boldsymbol{X}|} |\boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{X}| \right\} \tag{8-41}$$

那么，如果 $\theta<\varphi$，就说明 $\boldsymbol{X}$ 所代表的区域更接近于线特性而不是点特性。

如果考虑 3×3 的模板，则问题就成为 9 维的，但前边讨论的概念仍然适用。这里，需要 9 个 9 维正交矢量形成一个完整的基。这 9 个模板如图 8-8 所示。其中前 4 块模板(a)、(b)、(c)、(d)适合于边缘检测；(e)、(f)、(g)、(h)4 块模板适合于检测线；最后一块模板(i)则正比于一幅图像中模板所在区域的像素平均值。



1	$\sqrt{2}$	1
0	0	0
1	$-\sqrt{2}$	−1

(a)

1	0	−1
$\sqrt{2}$	0	$-\sqrt{2}$
1	0	−1

(b)

0	−1	$\sqrt{2}$
1	0	−1
$-\sqrt{2}$	1	0

(c)

$\sqrt{2}$	1	0
−1	0	1
0	1	$-\sqrt{2}$

(d)

0	1	0
−1	0	−1
0	1	0

(e)

−1	0	1
0	0	0
1	0	−1

(f)

1	−2	1
−2	4	−2
1	−2	1

(g)

−2	1	−2
1	4	1
−2	1	−2

(h)

1	1	1
1	1	1
1	1	1

(i)

图 8-8　正交模板

如果由 $\boldsymbol{X}$ 代表的 3×3 区域，并假定矢量 $\boldsymbol{W}_i$，$i=1,2,\cdots,9$ 是归一化的，从前边的讨论有：

$$p_e = \left[\sum_{i=1}^{4} (\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X})^2 \right]^{\frac{1}{2}} \tag{8-42}$$

$$p_l = \left[\sum_{i=5}^{8} (\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X})^2 \right]^{\frac{1}{2}} \tag{8-43}$$

$$p_a = |(\boldsymbol{W}_9^{\mathrm{T}}\boldsymbol{X})| \tag{8-44}$$

式中 p_e、p_l、p_a 分别是 $\boldsymbol{X}$ 向边缘、线和平均子空间投影的幅度。同样道理，有

$$\theta_e = \arccos\left\{ \frac{1}{|\boldsymbol{X}|}\left[\sum_{i=1}^{4} (\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X})^2 \right]^{\frac{1}{2}} \right\} \tag{8-45}$$

$$\theta_l = \arccos\left\{ \frac{1}{|\boldsymbol{X}|}\left[\sum_{i=5}^{8} (\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X})^2 \right]^{\frac{1}{2}} \right\} \tag{8-46}$$

$$\theta_a = \arccos\left\{\frac{1}{|X|}|W_9^{\mathrm{T}} X|\right\} \tag{8-47}$$

式中 θ_e，θ_l，θ_a 是 X 与它在边缘、线及平均子空间的投影之间的夹角。

采用这种检测方法可扩展到其他基与维数，只要基本矢量是正交的就可以。

8.1.3 区域生长

分割的目的是要把一幅图像划分成一些区域，对于这个问题的最直接的方法是把一幅图像分成满足某种判据的区域，也就是说，把点组成区域。为了实现分组，首先要确定区域的数目，其次要确定一个区域与其他区域相区别的特征，最后还要产生有意义分割的相似性判据。

分割区域的一种方法叫区域生长或区域生成。假定区域的数目以及在每个区域中单个点的位置已知，则可推导一种算法。从一个已知点开始，加上与已知点相似的邻近点形成一个区域。这个相似性准则可以是灰度级、彩色、组织、梯度或其他特性。相似性的测度可以由所确定的阈值来判定。它的方法是从满足检测准则的点开始，在各个方向上生长区域。当其邻近点满足检测准则就并入小块区域中，当新的点被合并后再用新的区域重复这一过程，直到没有可接受的邻近点时生成过程终止。

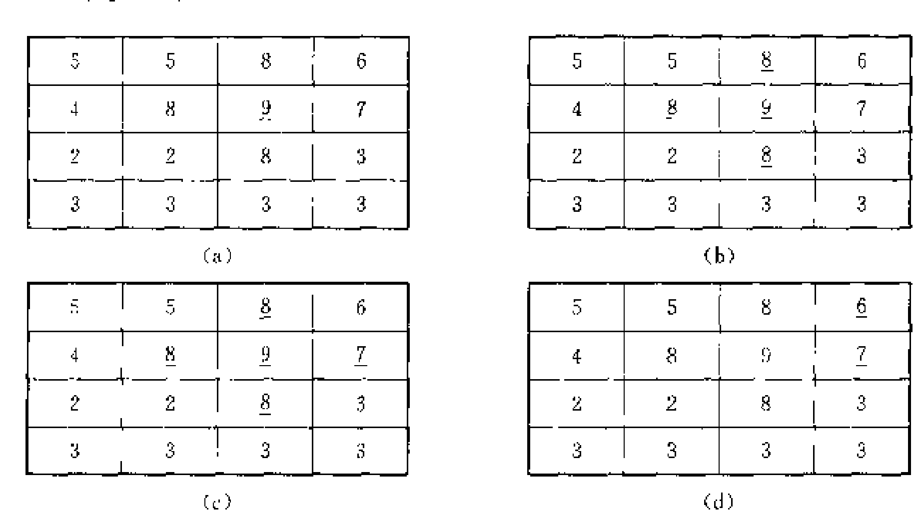


图 8-9 区域生长简例

图 8-9 示出一个简单的例子。这个例子的相似性准则是邻近点的灰度级与物体的平均灰度级的差小于 2。图中被接受的点和起始点均用一短线标出，其中(a)是输入图像；(b)是第一步接受的邻近点；(c)是第二步接受的邻近点；(d)是从 6 开始生成的结果。

当生成任意物体时，接受准则可以结构为基础，而不是以灰度级或对比度为基础。为了把候选的小群点包含在物体中，可以检测这些小群点，而不是检测单个点，如果它们的结构与物体的结构充分并且足够相似时就接受它们。另外，还可以使用界线检测对生成建立"势垒"，如果在"势垒"的近邻点和物体之间有界线，则不能把这邻近点接受为物体中的点。

8.1.4 区域聚合

区域聚合可直接用于图像分割。它要求聚合中的各个点必须在平面上相邻接而且特性相似。区域聚合的步骤是首先检查图像的测度集，以确定在测度空间中聚合的位置和数目，然后把这些聚合的定义用于图像，以得到区域聚合。一般区域聚合技术可以说明如下：

首先可以在图片上定义某个等价关系。例如，最简单的等价关系可定义为 $p(i,j)=p(k,$

l)。也就是说，如果 $p(i,j)=p(k,l)$，就说明 $p(i,j)$ 与 $p(k,l)$ 等价。任何在点的格子上的等价关系又可划分为等价类。例如 $p(i,j)$ 的取值范围为 0 到 63，就可以产生 64 个等价类的模板。如果关系满足，它的值等于 1，否则为 0。模板的图像是两两不相交的，那么 64 个模板就会充满整个格子。这些等价的类又可以进一步分为最大连接的子集，把这个叫做连接分量。连接性可以用点 (i,j) 的邻点来定义。如 4 连接邻点，8 连接邻点等等。4 连接邻点是 4 个非对角线上的 4 个邻点，8 连接则是环绕的 8 个邻点。假如 R 是属于格子的子集，在 R 中存在一个点序列，第一个点是 p_1，末一个点是 p_2，属于格子的子集 R 的两个点 p_1 和 p_2 是被连接起来的，这样，相继的各点是 4 连接相邻的。通过这样的连接关系可以定义一个属于 R 的子集，这个子集形成一个区域。在这个区域中，任何点都与 R 有关。利用等价模板可分成最大的连接区域，然后，这些最大的连接区域又可以像搭积木一样形成有意义的分割。

1970 年布赖斯和芬尼玛提出一种分割方法。这个方法如图 8-10 所示，图中(a)是具有灰度级的 3×3 的 $\boldsymbol{G}$ 阵列，图(b)是对 $\boldsymbol{S}$ 的分割结果。其中图像格子为 $\boldsymbol{G}$，它是大格子 $\boldsymbol{S}$ 的子格子。$\boldsymbol{G}$ 为 $n\times m$ 的格子，$\boldsymbol{S}$ 是 $(2n+1)\times(2m+1)$ 的大格子。在大格子中，$\boldsymbol{G}(i,j)$ 点位于 $\boldsymbol{S}$ 的 $(2i+1,2j+1)$ 点上。$\boldsymbol{G}$ 中的点与 $\boldsymbol{S}$ 中的点相对应，其中每一下标都是奇数，其余的点用来代表区域的边界。以这种形式表现的区域，产生一种寻找最大连接区域的方法。$\boldsymbol{G}$ 中的点与它上边和右边的点相比较，灰度级相同就合并，灰度级不同就插入边界线。把图像中的每个点都考虑了之后，整个图像就被分割成区域。在这个例子中，由于采用了 4 连接等价关系，因此，由图 8-10 可见，在对角线方向上的等灰度级产生了隔开的区域。

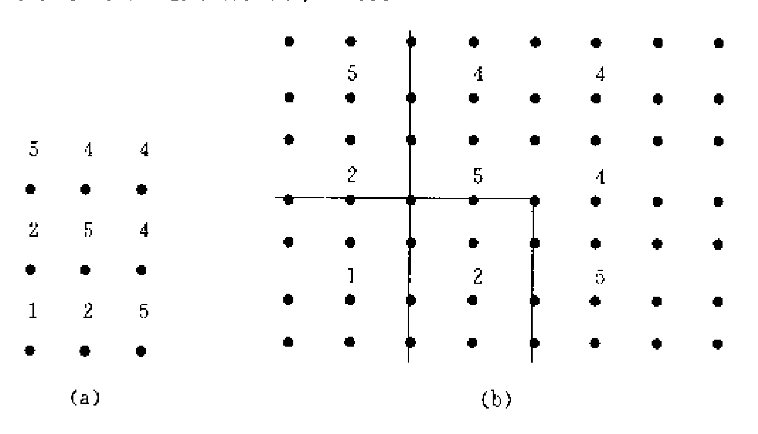


图 8-10 布赖斯和芬尼玛分割方法

8.2 描绘

当一幅图像被分割或确定之后，通常希望用一系列符号或某种规则来具体地描述该图像的特征，以便在进一步地识别、分析或分类中有利于区分不同性质的图像。同时，也可以减少图像区域中的原始数据量。一般把表征图像特征的一系列符号叫做描绘子。对这些描绘子的基本要求是它们对图像的大小、旋转、平移等变化不敏感。也就是说，只要图像内容不变，仅仅产生几何变化，描绘图像的描绘子将是惟一的。

8.2.1 区域描绘

1. 傅里叶描绘子

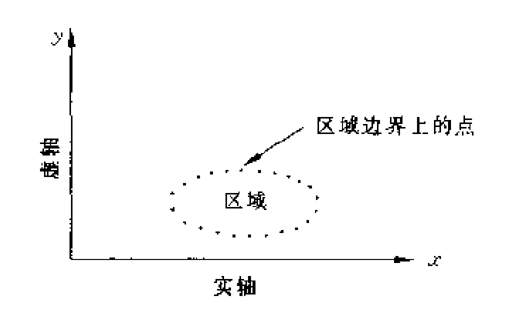


图 8-11　在复平面上区域边界的表示

当一个区域边界上的点已被确定时，可以从这些点中提取信息。这些信息就可以用来鉴别不同区域的形状。假如一个区域上有 M 个点可利用，可以把这个区域看作是在复平面内，纵坐标为虚轴，横坐标为实轴，如图 8-11 所示。

在边界上要分析每一个点的坐标(x,y)可以用一复数来表示，即：$x+jy$。从边界上任一点开始，沿此边界跟踪一周就可以得到一个复数序列。这个复数序列叫做傅里叶描绘子(FD)。

因为 DFT 是可逆的线性变换，因此，在这个过程中没有信息的增益或损失。对于形状的这种频域表示作些简单的处理就可以避免对于位置、大小及方向的依赖性。当给定了任意的 FD，用若干步骤可以使之归一化，从而不必考虑其原始形状的大小、位置及方向。

关于归一化问题可直接从 DFT 的性质中得出结论。例如，要改变轮廓大小，只要把 FD 分量乘一个常数就行了。由于傅里叶变换是线性的，它的反变换也会被乘以同样的常数。又如，把轮廓旋转一个角度，只要把每一个坐标乘以 $\exp(j\theta)$就可以使其旋转 θ 角。由 DFT 的性质，在空域旋转了 θ 角度，那么在频域中也会旋转 θ 角度。关于轮廓起始点的移动，由 DFT 的周期性可以看到，在空域中有限的数字序列实际上代表周期函数的一个周期。DFT 的系数就是这个周期函数的傅里叶表示式的系数。当轮廓的起点在空域中移动时，就相当于在频域中把第 k 次频率系数乘以 $\exp(jkT)$，这里 T 是周期的一部分，这部分即为起始点移动的部分。实际上这就是傅里叶变换的平移性质所导致的结果。当 T 从 $0\sim2\pi$ 变化时，则起点将把整个轮廓点经历一次。

给定一任意轮廓的 FD 后，归一化就可以执行一系列步骤，使轮廓有一个标准的大小、方向和起点。

在实际执行中还要考虑到如下一些问题：其一，如果取样不均匀将会给问题带来困难，因此，在理论上采用均匀间隔取样；其次，FFT 的算法要求阵列长度为 2 的整数次幂，这样在采用 FFT 之前，应调整表达式的长度。为作到这一点，首先计算出轮廓的周长，再用所希望的长度（当然应是 2 的整数次幂）去除，然后从一个起始点去追踪，所希望的 2 的幂次可以是大于序列长度的最小的 2 的幂次。

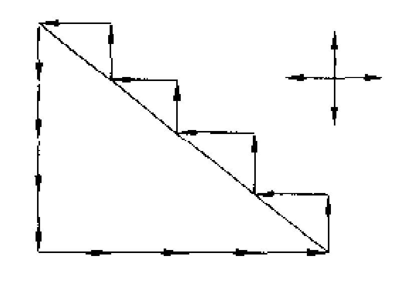


图 8-12　4 个取向的链码追迹

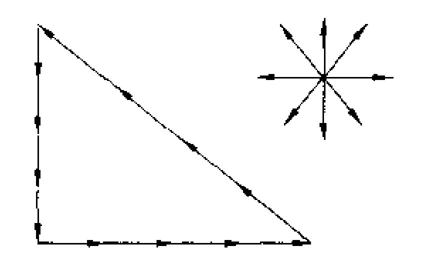


图 8-13　8 个取向的链码追迹

实际上，输入到形状分析运算中的将是从取样图片中取出的轮廓。这个轮廓的周长近似等于轮廓的实际周长。如果原始图像的取样密度足够高的话，那么序列将是轮廓的很好的近似。例如，图 8-12 所示的等边直角三角形，如果用 4 个取向的链码来追迹则会比较粗糙，如果用 8 个取向的链码来追迹，就得到精确得多的结果。由图 8-12 可见，追踪后的轮廓长度是直角边长

度的 4 倍。如图 8-13 所示，用 8 个方向来追踪，其结果更接近实际轮廓的长度。

在归一化中，为了克服噪声和量化误差带来的扰动，应选择最大幅度系数做为归一化系数。

图 8-14 是飞机侧影的描绘结果。这些结果是如下得到的：计算边界的 NFD(应用 512 点)；保留最低频率的 32 个点而把其他点置 0；求修改了的 512 阵列的傅里叶反变换，得到原始数据的近似。由图 8-14 所看到的结果，在最低的 32 个分量中的信息足以区别这些飞机的外形。

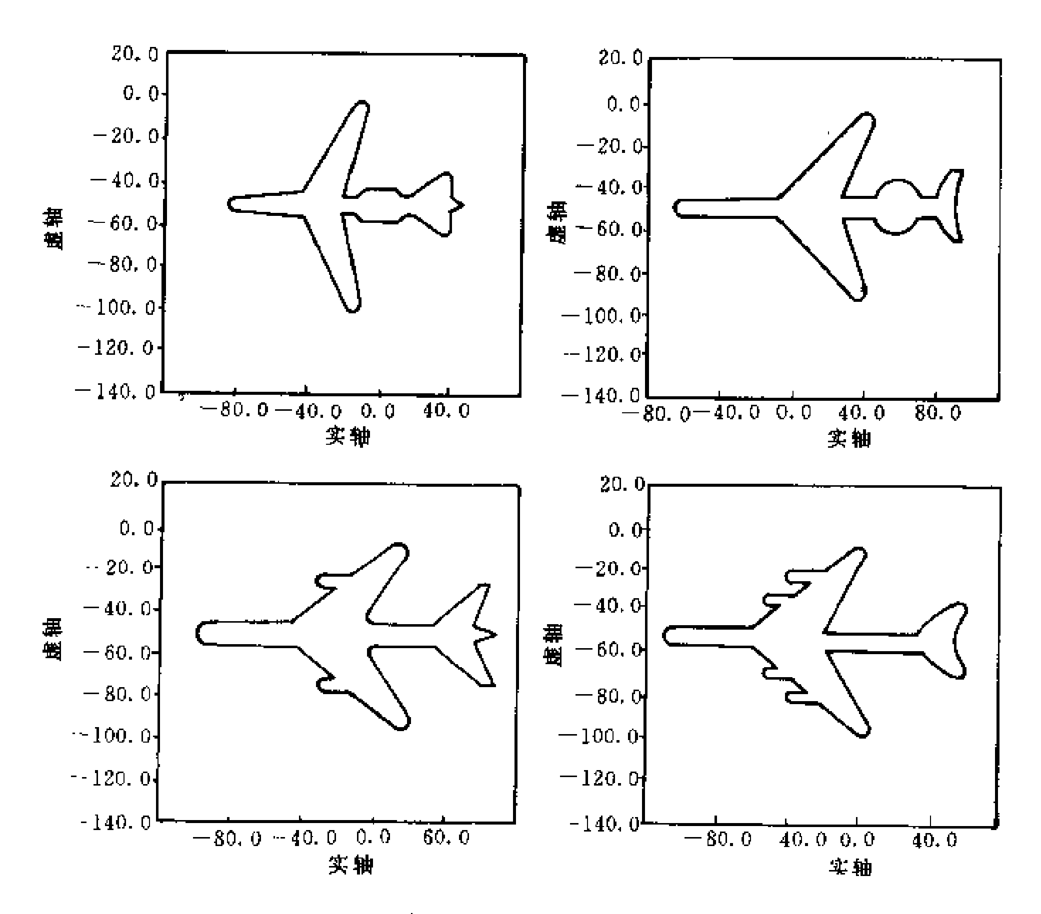


图 8-14 采用傅里叶描绘子得到的外形

2. 矩描绘子

采用傅里叶描绘子是以边界上的集合点(可用的)为基础。有时，一个区域以内部点的形式给出，那么，可用另外一种描绘子来描述。它对于图像的变换、旋转和大小变化都是恒定的，这就是矩描绘子。

设 $f(x,y)$ 是一个二维函数，可用下式来表示 $(p+q)$ 阶矩：

$$m_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p y^q f(x,y)\mathrm{d}x\mathrm{d}y \tag{8-48}$$

式中 $p,q=0,1,2,\cdots$。

中心矩由下式表示：

$$\mu_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\mathrm{d}x\mathrm{d}y \tag{8-49}$$

式中 $\bar{x}=\dfrac{m_{10}}{m_{00}}$, $\bar{y}=\dfrac{m_{01}}{m_{00}}$,对于数字图像来说,式(8-49)可表示为下式:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x,y) \tag{8-50}$$

m_{00}, m_{10}, m_{01} 如下:

$$m_{00} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\mathrm{d}x\mathrm{d}y \tag{8-51}$$

$$m_{10} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xf(x,y)\mathrm{d}x\mathrm{d}y \tag{8-52}$$

$$m_{01} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} yf(x,y)\mathrm{d}x\mathrm{d}y \tag{8-53}$$

如果假定所给图像 $f(x,y)$ 在每一点 (x,y) 处的灰度级是在 (x,y) 点的"质量",那么就可以定义 $f(x,y)$ 的重心点 $(\bar{x},\bar{y})$,其中

$$\bar{x} = \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xf(x,y)\mathrm{d}x\mathrm{d}y}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\mathrm{d}x\mathrm{d}y} \tag{8-54}$$

$$\bar{y} = \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} yf(x,y)\mathrm{d}x\mathrm{d}y}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\mathrm{d}x\mathrm{d}y} \tag{8-55}$$

因此有

$$\bar{x}=\frac{m_{10}}{m_{00}} \qquad \bar{y}=\frac{m_{01}}{m_{00}} \tag{8-56}$$

由上边各式可得到三阶中心矩如下:

$$\mu_{10} = \sum_x \sum_y (x - \bar{x})^1 (y - \bar{y})^0 f(x,y) = m_{10} - \frac{m_{10}}{m_{00}}(m_{00}) = 0 \tag{8-57}$$

$$\mu_{11} = \sum_x \sum_y (x - \bar{x})^1 (y - \bar{y})^1 f(x,y) = m_{11} - \frac{m_{10}m_{01}}{m_{00}} \tag{8-58}$$

$$\begin{aligned}\mu_{20} &= \sum_x \sum_y (x - \bar{x})^2 (y - \bar{y})^0 f(x,y) \\ &= m_{20} - \frac{2m_{10}^2}{m_{00}} + \frac{m_{10}^2}{m_{00}} = m_{20} - \frac{m_{10}^2}{m_{00}}\end{aligned} \tag{8-59}$$

$$\mu_{02} = \sum_x \sum_y (x - \bar{x})^0 (y - \bar{y})^2 f(x,y) = m_{02} - \frac{m_{01}}{m_{00}} \tag{8-60}$$

$$\mu_{30} = \sum_x \sum_y (x - \bar{x})^3 (y - \bar{y})^0 f(x,y) = m_{30} - 3\bar{x}m_{20} + 2m_{10}\bar{x}^2 \tag{8-61}$$

$$\mu_{12} = \sum_x \sum_y (x - \bar{x})^1 (y - \bar{y})^2 f(x,y) = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \tag{8-62}$$

$$\mu_{21} = \sum_x \sum_y (x - \bar{x})^2 (y - \bar{y})^1 f(x,y) = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01} \tag{8-63}$$

$$\mu_{03} = \sum_x \sum_y (x - \bar{x})^0 (y - \bar{y})^3 f(x,y) = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01} \tag{8-64}$$

概括起来有如下一些结果:

$$\mu_{00} = m_{00} \qquad \mu_{01} = 0 \qquad \mu_{10} = 0$$

$$\mu_{20} = m_{20} - \bar{x}m_{10} \qquad \mu_{02} = m_{02} - \bar{y}m_{01}$$
$$\mu_{11} = m_{11} - \bar{y}m_{10}$$
$$\mu_{30} = m_{30} - 3\bar{x}m_{20} + 2m_{10}\bar{x}^2$$
$$\mu_{12} = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2m_{10}$$
$$\mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2m_{01}$$
$$\mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2m_{01}$$

定义归一化中心矩为

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{8-65}$$

$$\gamma = \frac{p+q}{2} \tag{8-66}$$

利用第二阶和第三阶矩可导出 7 个不变矩组：

$$\phi_1 = \eta_{20} + \eta_{02}$$
$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + 3\eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{12} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\phi_7 = (3\eta_{12} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{12} + \eta_{03})^2] \tag{8-67}$$

Hu(1962)已经证明了这个矩组对于平移、旋转和大小比例变化都是不变的，因此用它们可以描绘一幅给定的图像。

3. 拓扑描绘子

拓扑学是研究图形性质的理论。拓扑特性可用于描绘图像平面区域。有些图形只要不撕裂或连接，其拓扑性质并不受形变的影响。图 8-15 是带有两个孔的图形，如果把区域中孔洞数做为拓扑描绘子，显然这个性质不受伸长或旋转变换的影响，但是，如果撕裂或折叠时孔洞数就要变化了。

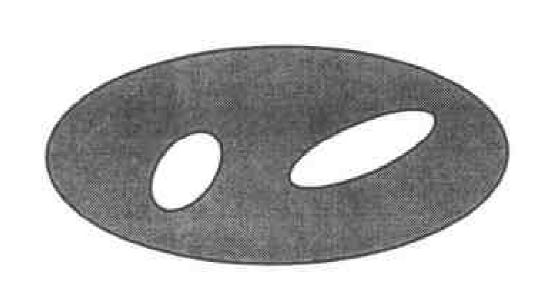


图 8-15　带有两个空洞的区域

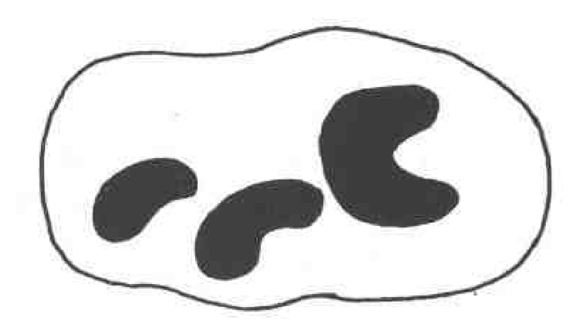


图 8-16　有三个连接部分的区域

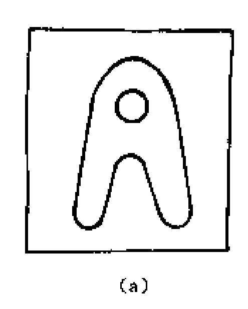
区域描绘的另一种有用的拓扑特性是连接部分的个数。一个集合的连接部分就是它的最

大子集，在这个子集中的任何两点都可以用一条完全在子集中的曲线加以连接。图 8-16 所示图形就有三个连接部分。

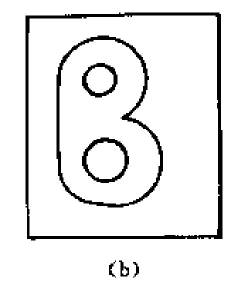
如果一幅图像的孔洞数为 H，连接部分为 C，则欧拉数的定义如式(8-68)所示。欧拉数

$$E = C - H \tag{8-68}$$

也是拓扑特性之一。如图 8-17(a)所示图形有一个连接部分和一个孔，所以它的欧拉数为 0；而图(b)有一个连接部分和二个孔，所以它的欧拉数为 -1。



(a)



(b)

图 8-17　具有欧拉数为 0 和 -1 的图形

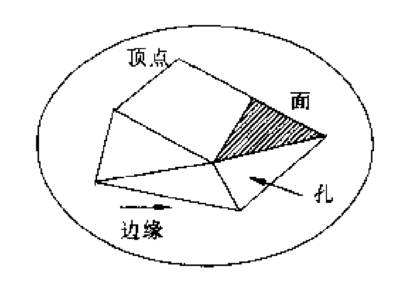


图 8-18　包含多角网络的区域

由直线表示的区域，按照欧拉数有一个简单的解释。如图 8-18 所示的多角网络，把这样的网络内部区域分成面和孔。如果设顶点数为 W，边缘数为 Q，面数为 F，将得到下列关系，这个关系称为欧拉公式：

$$W - Q + F = C - H \tag{8-69}$$

即：

$$W - Q + F - C - H = E$$

在图 8-18 的多角网络中，有 7 个顶点、11 条边、2 个面、1 个连接区、3 个孔，因此，由式(8-69)可得到：$7-11+2=1-3=-2$。

拓扑的概念通常在图像中确定特征区域很有用。

8.2.2　关系描绘

如果图像已经被分割为区域或部分，则图像描绘的下一步任务就是如何把这些元素组织成为有意义的关系结构。结构描绘一般是以文法概念为基础的。例如，从一幅图像中已分割出图 8-19 所示的阶梯形结构，要用某种方法来描绘它，首先要定义一些基本元素，然后再定义一个重写规则就可以描绘出此阶梯形结构。图中(a)是阶梯结构，(b)是基本元素，(c)是编码结构。

在描绘过程中规定基本元素为 a 和 b，重写规则如下：

$$S \rightarrow aA$$

$$A \rightarrow bS$$

$$A \rightarrow b$$

这里 S 和 A 是变量，元素 a 和 b 是常量。第一个规则说明 S 可以用基本元素 a 和变量 A 来代替，变量 A 可以用 b 和 S 来代替，也可以用 b 来代替。如果用 bS 来代替 A，则可以重复第一个规则的步骤。如果用 b 来代替，则步骤终止。这里假定都用 S 为起始点，第一个元素后面总是 b。由上例可见只需三条重写规则就可以产生无穷多的相似结构。

1. 串文法和语言

图 8-19 说明的编码结构是由符号的连接串组成的。这种符号串可以引用形式语言的概念来处理。形式语言起源于 1950 年，诺姆、乔姆斯基用数学模型研究了文法，目的是研究一种计

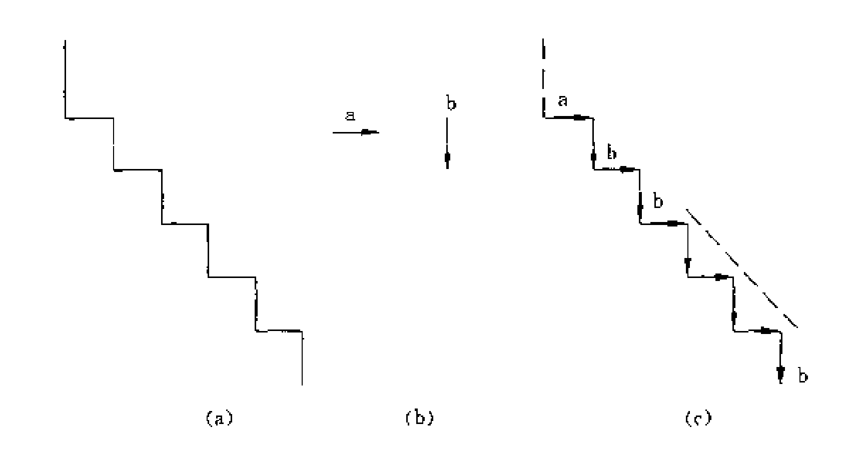


图 8-19 阶梯形结构之描绘

算机文法，然后用这个文法去研究自然语言，以便计算机在翻译和解答问题的过程中解释自然语言。关于形式语言的研究和应用已渗透到编译设计、计算机语言、自动机理论及模式识别和图像处理领域中。

首先，介绍一些最基本的定义。

定义 V 为任何有限的符号集合；在 V 范围内的一个句子、一串字符或字都是由集合中的符号组成的任何有限长度的串。例如，给定 $V=\{0,1\}$，则 $\{0,1,00,01,11,000,001,\cdots\}$ 都是有效的句子。定义没有符号的句子为空句子，用 λ 来代表。用 V^* 代表由 V 中的符号组成的所有句子集合，其中包括空句子。V^+ 代表 $(V^*-\lambda)$ 的句子集合。

例如，字母 $V=\{a,b\}$，$V^*=\{\lambda,a,b,ab,aa,ba,\cdots\}$，$V^+=\{a,b,ab,aa,ba,\cdots\}$ 语言都是在字母范围内句子的任意集合。

形式语言理论主要研究文法及其性质。串文法（或叫简单文法）是四元的，即

$$G=(V_N,V_T,P,S)$$

其中：V_N 为非终端符集合（变量）；V_T 为终端符的集合（常量）；P 为产生式或重写规则集合；S 为起始符或根符号。假定 S 属于集合 V_N，并且 V_N 和 V_T 是不相交的集合，字母 V 是 V_N 和 V_T 的合集。

由形式串文法 G 产生的语言记作 $L(G)$，这个语言就代表了一个模式。由字符产生的语言 $L(G)$ 满足两个条件：每一串只由终端符组成，每一串都由 S 开始并用由 P 决定的产生式来生成。

例如，有文法 $G=(V_N,V_T,P,S)$，$V_N=\{S\}$，$V_T=\{a,b\}$，$P=\{S\to aSb,S\to ab\}$，如果把第一个产生式用 $m-1$ 次，然后再用第二个产生式，由此可产生下列语言：

$$S\to aSb\to aaSbb\to a^2Sb^2\to a^{m-1}Sb^{m-1}\to a^mb^m$$

显然，这个文法产生的语言看作仅仅由这种形式的串组成，特定的串长取决于 m，m 可以是任意整数。可以把 $L(G)$ 表达为下面的形式：

$$L(G)=\{a^mb^m|_{m\geq 1}\}$$

这个简单的文法可以用来产生无限多串组成的语言。

2. 文法的类型

在讨论文法类型之前，为了便于叙述，规定所使用的符号如下：非终端符 V_N 用大写英文字母表示，如 $S,A,B,C,\cdots$；终端符 V_T 用小写英文字母表示，如 $a,b,c,\cdots$；终端字符串用英文字母表后边的小写字母来表示，如 $v,w,x,y,\cdots$；终端和非终端的混合字符串用小写的希腊字

母来表示，如 $\alpha,\beta,\gamma,\delta,\cdots$。

我们把产生式的一般形式为 $\alpha\rightarrow\beta$ 的文法叫短语结构文法，一般根据加于产生式的约束条件而把文法分成如下几类：

(1) 无约束文法

这种文法的产生式为 $P:\alpha\rightarrow\beta$，其中箭头的左右端可以是任意形式的链。

(2) 上下文有关的文法

这种文法的产生式为 $P:\alpha_1A\alpha_2\rightarrow\alpha_1\beta\alpha_2$，其中 $A\in V_N$，α_1,α_2，$\beta\in V^*$，$\beta\neq\lambda$，这种文法规定只有当 A 出现在 α_1 和 α_2 串的前后关系 $\alpha_1A\alpha_2$ 中时，才允许用串 β 来代替非终端符 A。

(3) 上下文无关的文法

此文法的产生式为 $P:A\rightarrow\beta$，其中 $A\in V_N$，$\beta\in V^*$。这个文法并不考虑出现 A 的上下文就可以用串 β 去代替 A 变量。

(4) 正则文法

也称为有限状态文法。它的产生式为 $P:A\rightarrow\alpha B$ 或 $A\rightarrow\alpha$，这里 A，$B\in V_N$，$a\in V_T$，A、B、α 均是单个符号。还可以选择另外一种产生式，即 $P:A\rightarrow B\alpha$ 和 $A\rightarrow\alpha$。当然二种产生式中只能选一种，而不可同时选用。

上述四类文法有时依次称为 0 型，1 型，2 型和 3 型文法。由它们产生的语言分别称为 0 类型语言，1 类型语言，2 类型语言和 3 类型语言。由上述分类可以看出，所有的正则文法都是上下文无关的，所有上下文无关的文法都是上下文有关的，所有上下文有关的文法都是无约束的。对上述四类文法下面分别举例说明。

例 1　无约束文法。

$$G=(V_N,V_T,P,S)$$
$$V_N=\{A,B,S\}$$
$$V_T=\{a,b,c\}$$

P: ① $S\rightarrow aAbc$

② $Ab\rightarrow bA$

③ $Ac\rightarrow Bbcc$

④ $bB\rightarrow Bb$

⑤ $aB\rightarrow aaA$

⑥ $aB\rightarrow a$

$$S\xrightarrow{①}aAbc\xrightarrow{②}abAc\xrightarrow{③}abBbcc\xrightarrow{④}aBbbcc\xrightarrow{⑤}aaAbbcc\xrightarrow{②}aabAbcc\xrightarrow{②}aabbAcc$$
$$\xrightarrow{③}a^2b^2Bbccc\xrightarrow{④}a^2bBbbccc\xrightarrow{④}a^2Bbbbccc\xrightarrow{⑥}a^2b^3c^3$$

所以有
$$L(G)=\{a^nb^{n+1}c^{n+1}\}_{n\geq 0}$$

例 2　上下文有关的文法。

$$G=(V_N,V_T,P,S)$$
$$V_N=\{A,B,S\}$$
$$V_T=\{a,b,c$$

P: ① $S\rightarrow abc$

② $S\rightarrow aAbc$

③ $Ab\rightarrow bA$

$$④\ Ac \rightarrow Bbcc$$

$$⑤\ bB \rightarrow Bb$$

$$⑥\ aB \rightarrow aaA$$

$$⑦\ aB \rightarrow aa$$

$$S \xrightarrow{②} aAbc \xrightarrow{③} abAc \xrightarrow{④} abBbcc \xrightarrow{⑤} aBbbcc \xrightarrow{⑦} aabbcc \longrightarrow a^2b^2c^2$$

所以
$$L(G)=(a^nb^nc^n|_{n\geq 1})$$

例 3　上下文无关的文法。

$$G=(V_N,V_T,P,S)$$

$$V_N=\{S\}$$

$$V_T=\{a,b,c\}$$

$$P:①\ S\rightarrow ab$$

$$②\ S\rightarrow aSb$$

$$S \xrightarrow{②} aSb \xrightarrow{②} aaSbb \xrightarrow{①} aaabbb \longrightarrow a^3b^3$$

所以
$$L(G)=(a^nb^n|_{n\geq 1})$$

例 4　正则文法。

$$G=(V_N,V_T,P,S)$$

$$V_N=\{S\}$$

$$V_T=\{a,b\}$$

$$P:①\ S\rightarrow a$$

$$②\ S\rightarrow b$$

$$③\ S\rightarrow aS$$

$$④\ S\rightarrow bS$$

$$⑤\ A\rightarrow aA$$

$$⑥\ A\rightarrow a$$

$$S \xrightarrow{③} aS \xrightarrow{③} aaS \xrightarrow{③} aaaS \xrightarrow{①} a^4$$

$$S \xrightarrow{④} bS \xrightarrow{④} bbS \xrightarrow{②} b^3$$

$$S \xrightarrow{③} aS \xrightarrow{④} abS \longrightarrow \cdots \longrightarrow a^nb^m$$

所以
$$L(G)=a^nb^m|_{m,n>0}$$

3. 位置算子的运用

前述的字符串是一维结构，而图像是二维结构，因此，在用字符串来描绘图像时就需要建立一种相应的方法，把二维的位置关系缩减为一维形式。

串文法在图像描绘中大多数是从物体中抽取的联接线段为基础的。这种方法如图 8-20 所示。这是用有特定方向和长度的线段把结果编码。

另外一种方法如图 8-21 所示。利用有向线段并用已定义的一些运算来描绘图像的某些部分。图中(a)是用有向线段来表示某些区域，图(b)是定义的某些运算。下面用一个具体例子来说明这些概念。

例 1　图片描绘语言。

$$G=\{V_N,V_T,P,S\}$$

$$V_N = \{S, A_1, A_2, A_3, A_4, A_5\}$$

$$V_T = \{a\nearrow, b\searrow, c\rightarrow, d\downarrow\}$$

P：① $S \rightarrow d + A_1$

② $A_1 \rightarrow c + A_2$

③ $A_2 \rightarrow \sim d * A_3$

④ $A_3 \rightarrow a + A_4$

⑤ $A_4 \rightarrow b * A_5$

⑥ $A_5 \rightarrow c$

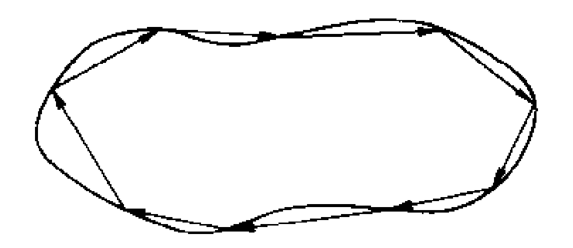


图 8-20　用有方向的线段描绘区域边界

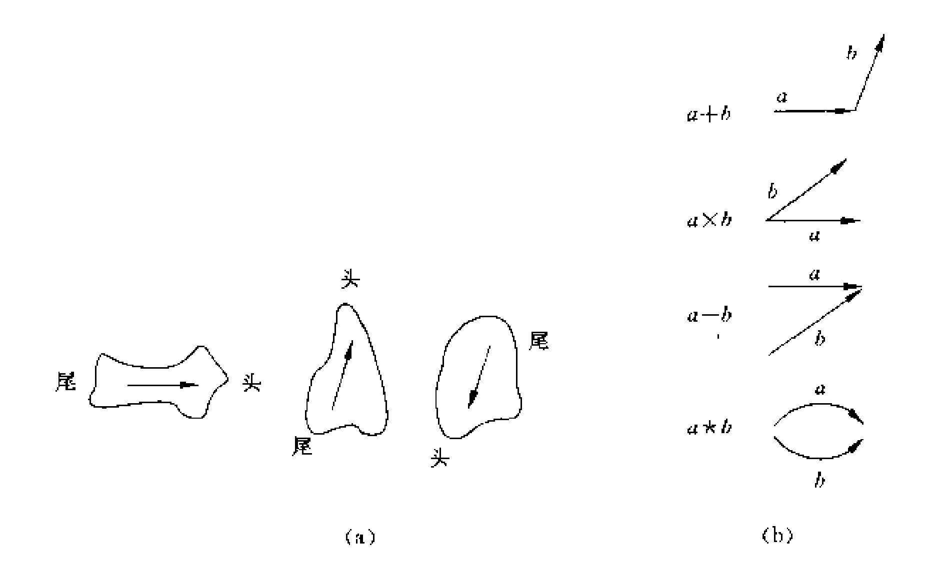


图 8-21　另一种描绘方法

这里($\sim d$)表示与基本单元d方向相反的像元，而a、b、c、d的方向如V_T中所定义的那样。应用产生式产生一个像元d，其后，跟随一个尚未定义的变量A_1。但是，A_1分量所表示的结构的尾在这点上将和d的头相连，这是由规定的算法"+"所决定的。变量A_1又可分解为$c+A_2$，当然A_2尚未定义。同样A_2又可分解为$\sim d * A_3$。应用前三条产生式得到的结果如图 8-22 中(a)、(b)、(c)所示。算子"*"定义为尾到尾、头到头的连接方式。使用全部产生式所得到的最后结果示于图 8-22(f)中。这种 PDL 文法只能产生一个结构。如果在产生式的规则中引入递归规则(变量有代替自己的能力)，则这种文法所产生的结构可扩展到各种结果。

例 2　利用例 1 的条件，定义下列产生式规则：

① $S \rightarrow d + A_1$

② $A_1 \rightarrow c + A_1$

③ $A_1 \rightarrow \sim d * A_2$

④ $A_2 \to a + A_2$

⑤ $A_2 \to b * A_2$

⑥ $A_2 \to c$

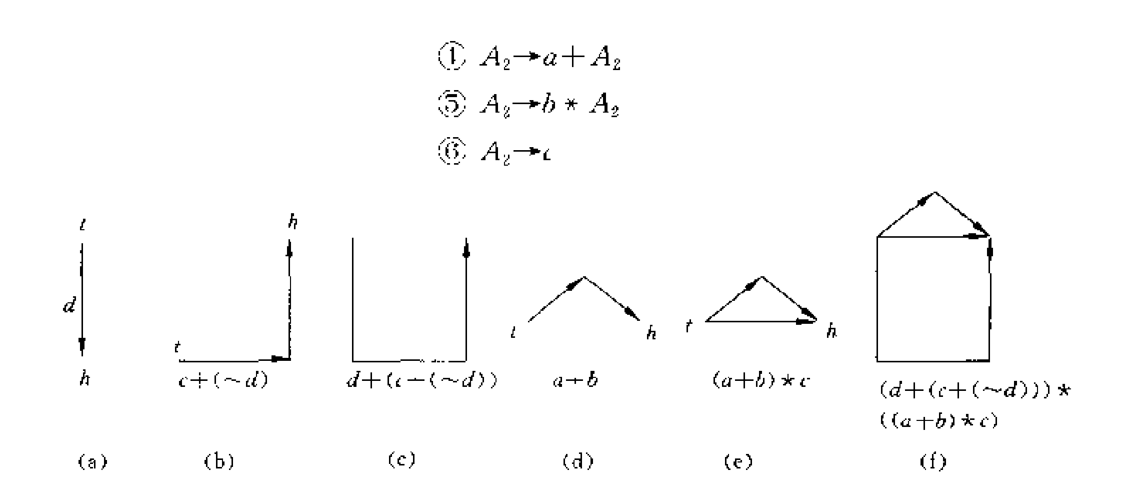


图 8-22　PDL 结构组成步骤

如果顺次应用这些产生式就会产生图 8-22(f)的结果。

例 3　染色体特征的描绘

描绘染色体的文法所用的基本像元如图 8-23(a)所示。这些基本像元是对染色体沿边界顺时针方向循迹而检测出来的。典型的半中期和末期染色体的形状如图 8-23(b)所示，描绘文法如下：

$$G=(V_N, V_T, P, S)$$

$$V_T=\{a,b,c,d,e\}$$

$$V_N=\{S,T,A,B,C,D,E,F\}$$

P：① $S \to C \cdot C$

② $T \to A \cdot C$

③ $C \to B \cdot C$

④ $C \to C \cdot B$

⑤ $C \to F \cdot D$

⑥ $C \to E \cdot F$

⑦ $E \to F \cdot c$

⑧ $D \to c \cdot F$

⑨ $A \to b \cdot A$

⑩ $A \to A \cdot b$

⑪ $A \to e$

⑫ $B \to b \cdot B$

⑬ $B \to B \cdot b$

⑭ $B \to b$

⑮ $B \to d$

⑯ $F \to b \cdot F$

⑰ $F \to F \cdot b$

⑱ $F \to a$

其中算子"·"用于描绘在按顺时针方向追迹边界时在产生式中各项的可连接性。在这个文法中，S 和 T 均为起始符，S 可产生相应于半中期染色体的结构，T 可以产生相应于末期染色

体的结构。

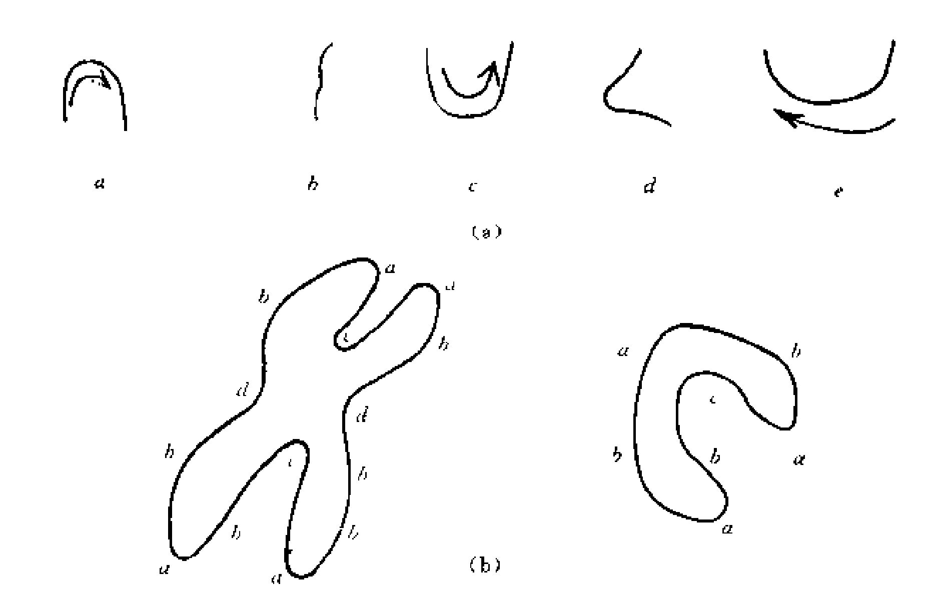


图 8-23　用边界轨迹描述染色体

4. 高维文法

串文法适用于那些图像元素的连接可以用从头到尾或用其他连续形式的图像元素的描绘。在这里我们考虑一种更普遍的文法描绘途径，它有能力描绘更高级的图像元素。

(1) 树文法

高维文法之一是所谓树文法。树文法中所定义的树是一个或一个以上的节点的集合。其中有一个惟一指定的节点为根；剩下的节点划分为 m 个不相交的集合，这些集合为 $T_1, T_2, \cdots, T_m$，把 T_m 叫做 T 的子树，树尖是树的根干部节点的集合，取从左到右的次序。图 8-24 是一树图，其中 \$ 是根(root)，x、y 为树尖。

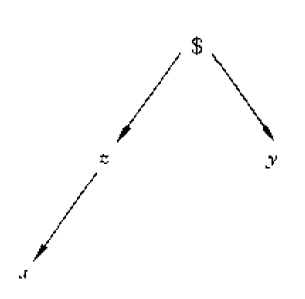


图 8-24　树图

一般来讲，在树图中有两类重要信息。一是关于节点的信息；另一个便是节点与其邻点的有关信息。在存储时，节点是用一组字描述并存储的，而节点与邻点的有关信息是以对其邻点的指示符的集合的形式存储。第一类信息用于识别模式的像元，第二类信息定义像元和其他子结构间的物理关系。例如，把图 8-25 所示的关系用图(b)所示的树来表示。图 8-25(b)中，\$ 表示根，在 \$ 中包含着 a 和 c 两部分，因此，从根放射出两个分支。第二级，a 中包含 b，c 中包含 d 和 e，在 e 中又包含 f。这样就构成了一个树图，其中 b，d，f 是树的叶子。

树文法为五元式，即

$$G = (V_N, V_T, P, r, S)$$

其中：V_N 为非终端符；V_T 为终端符；S 为起始符；P 为产生式集合；r 为秩函数，它指出一个节点直接下降的数目。

例　利用树文法把图 8-26(a)的电路结构表示成树的形式。这个树图是把 L-C 网络的最左边的节点定义为根。其树文法如下(电路中的 L 用 l、C 用 c)：

$$G = (V_N, V_T, P, r, S)$$

$$V_N = \{S, A\}$$

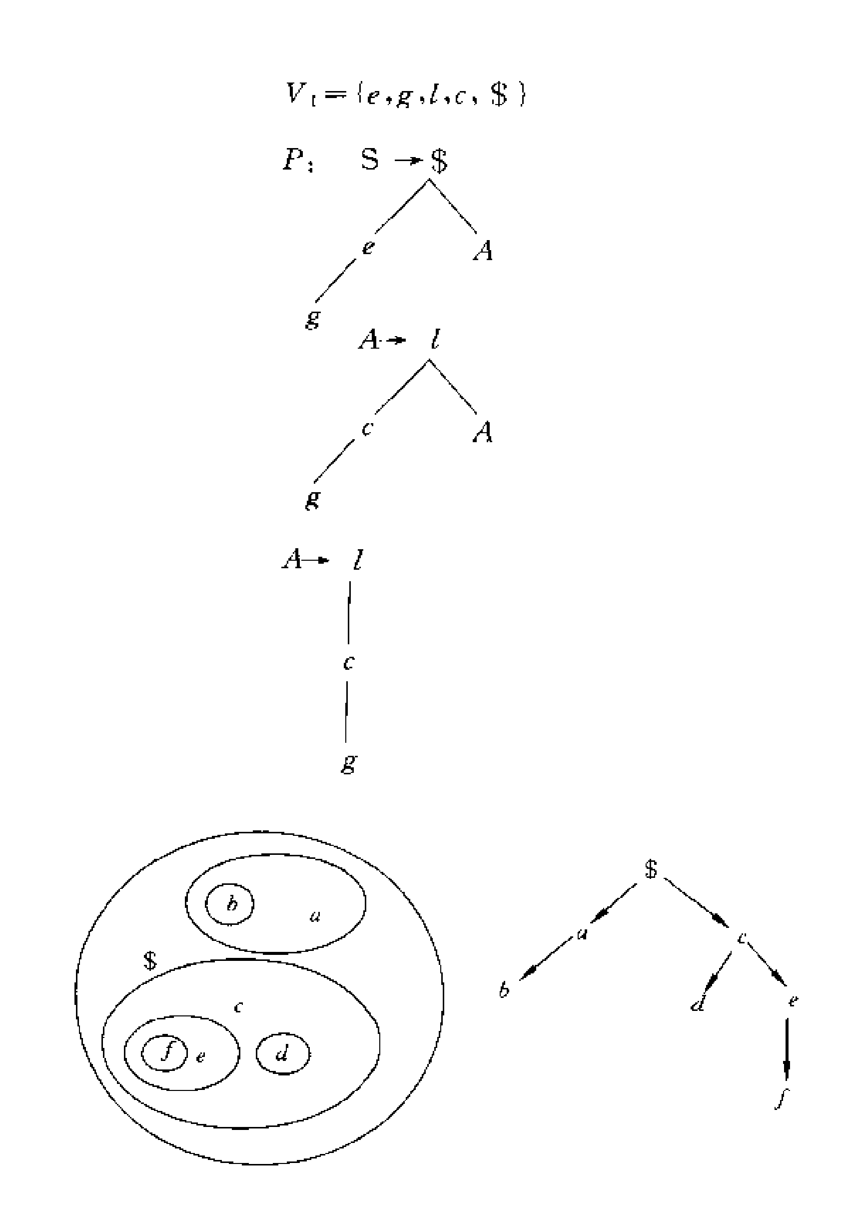


图 8-25 用树来表示简单的组合区域

在这种情况下，秩函数 $r(e)=1$, $r(g)=0$, $r(l)=\{2,1\}$, $r(c)=1$, $r(\$)=\{2,1\}$利用 3 条产生式并利用 A 上定义的递归性就能产生无限数量的结构。

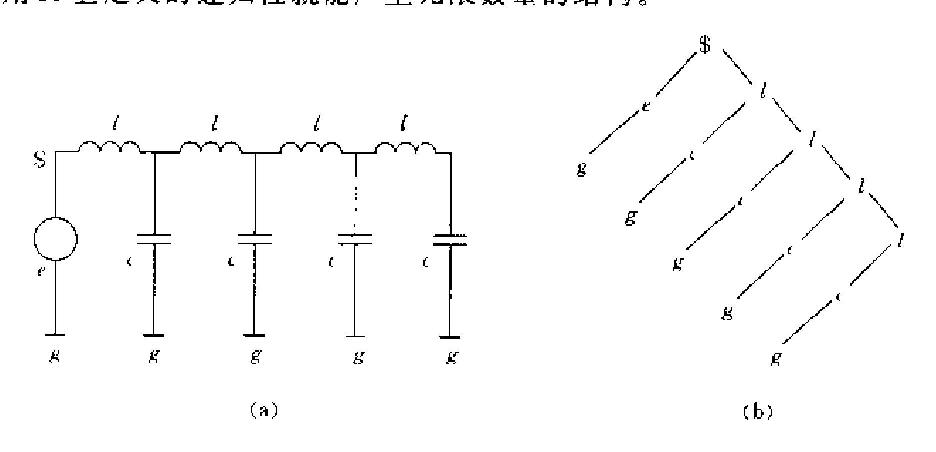


图 8-26 图形的树表示

(2) 网文法

网是把节点加以标号的无指向图结构。在作图像描绘时，它的表示法比串或树更加简单。图 8-27 示出了一些简单的网。

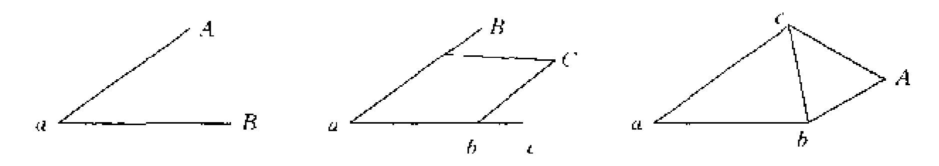


图 8-27 一些简单的网

短语结构串文法的重写规则比较简单，但是包含有网的重写规则就比较复杂了。如果要用另外一个子网 β 代替 ω 网中的子网 α，就需要规定如何把 β 嵌入 ω 以代替 α。为此，要规定嵌入规则。但是，如果想要能够在任何包含 α 做为子网的网中用 β 代替 α，就要使嵌入规则的定义与主网 ω 无关。

设 V 是标记的集合，N_α 和 N_β 分别是 α 和 β 网的节点的集合。可定义一个三重网的重写规则 (α,β,ϕ)，其中 ϕ 是从 $N_\alpha\times N_\beta$ 映射到 2^V 的函数，这个函数规定了嵌入 β 代替 α 的规则，即规定如何把 β 的节点连接到被移走的子网 α 的每一个节点的邻点处。$N_\alpha\times N_\beta$ 是指集合和的笛卡儿乘积，也就是取序集对 (n,m) 以便使 n 是 N_β 中的元素，m 是 N_α 中的元素。ϕ 是序集对 $N_\beta\times N_\alpha$ 的函数，其宗量取 (n,m) 的形式，此处的 n 在 N_β 中，m 在 N_α 中。$\phi(n,m)$ 的值规定了允许把 n 连接到 m 的邻点上。例如，$\phi(A,B)=\{C,D\}$ 的意思是把 β 中的节点 B 连接到节点 A 的邻点上，A 在 α 中。它们的标记或者是 C，或者是 D。

网文法定义为四元式，即：

$$G=(V_N,V_T,P,S)$$

其中：V_N 为非终端词汇；V_T 为终端词汇；P 为网的产生式规则的集合；S 为起始符号。一般来讲，S 在 V_N 内，词汇 V 是 V_N 和 V_T 的集合。

例如，网文法 $G=(V_N,V_T,P,S)$

$V_N=\{S\}$

$V_T=\{a,b,c\}$

P： α β ϕ

① $\dot{S}$ b, a, S, c $\phi(a,S)=\{b,c\}$

② S b, a, c $\phi(a,S)=\{b,c\}$

这里用 ϕ 说明的嵌入指出，把 β 的节点 a 连接到 S 的标号为 b 和 c 的邻点上，a 可用 β 来重写。但是，这个规则不能应用于第一次产生式执行的情况，因为一个单点网的起始没有邻接点。在这样的情况下，可以理解为第一次运用产生式时 ϕ 为零。这种网文法可以产生图 8-28 所示的结构。

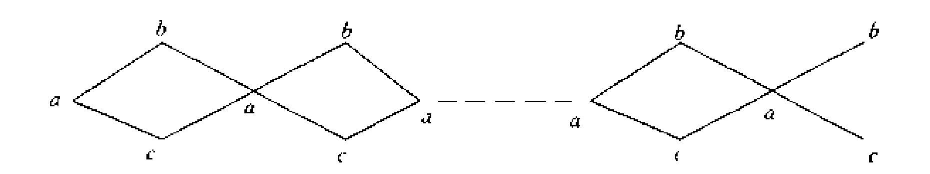


图 8-28　由网文法产生的结构

上面定义的网文法类似于无约束串文法。如果对产生式加以限制，就可以定义出约束型网文法。

如果存在一个 α 的非终端点 A，使得($\alpha-\beta$)是 β 的一个子网，网的重写规则(α,β,ϕ)叫做上下文有关的。如果 α 包含一个单点，把(α,β,ϕ)叫做上下文无关的。

网文法的一种特殊情况是网中的终止符集 V_T 仅由一个符号组成。在这种情况下，由文法产生的每一个网的每一个点具有相同的标号，于是可以忽略标号而以其基本图来识别网，这种类型的网文法被称为"图文法"。

例 1　考虑上下文有关图文法如下：

$$G=(V_N,V_T,P,S)$$
$$V_N=\{A,B,C,S\}$$
$$V_T=\{a\}$$

P:	α	β	ϕ
①	S	A•—• •B	$\phi(A,S)=\{\alpha\}$
②	S	A•—• •B	$\phi(A,B)=\{A,B\}$
③	B	A, C, α	$\phi(A,B)=\{A,\alpha\}$
④	C, α	α•—•C, α•—•α	$\phi(\alpha,C)=\{A,\alpha\}$ $\phi(\alpha,\alpha)=\{A,\alpha\}$
⑤	C, α	α, α, •S	$\phi(\alpha,C)=\{A,\alpha\}$ $\phi(\alpha,\alpha)=\{A,\alpha\}$
⑥	$A\mid B\mid C$	α	正则

这个文法产生的图形结构包含有任何数量的串联和并联部分。并联的分段至少被一个串联节隔开，并且所有的结构的起始和终端至少有一个这样的节。两个简单的结果如图 8-29 所示。

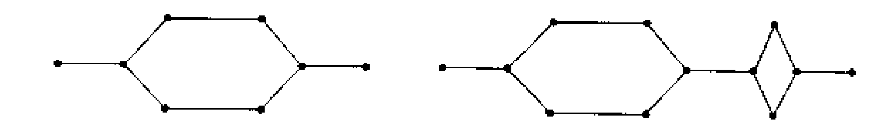
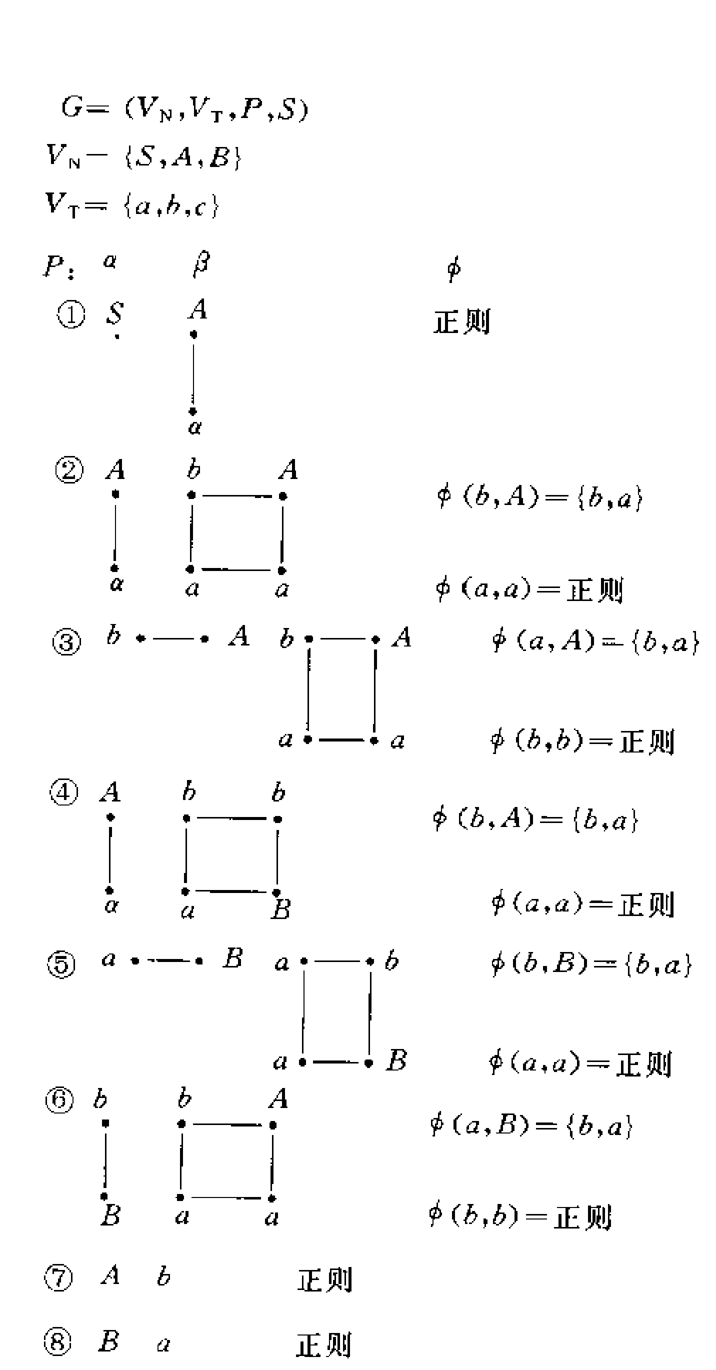
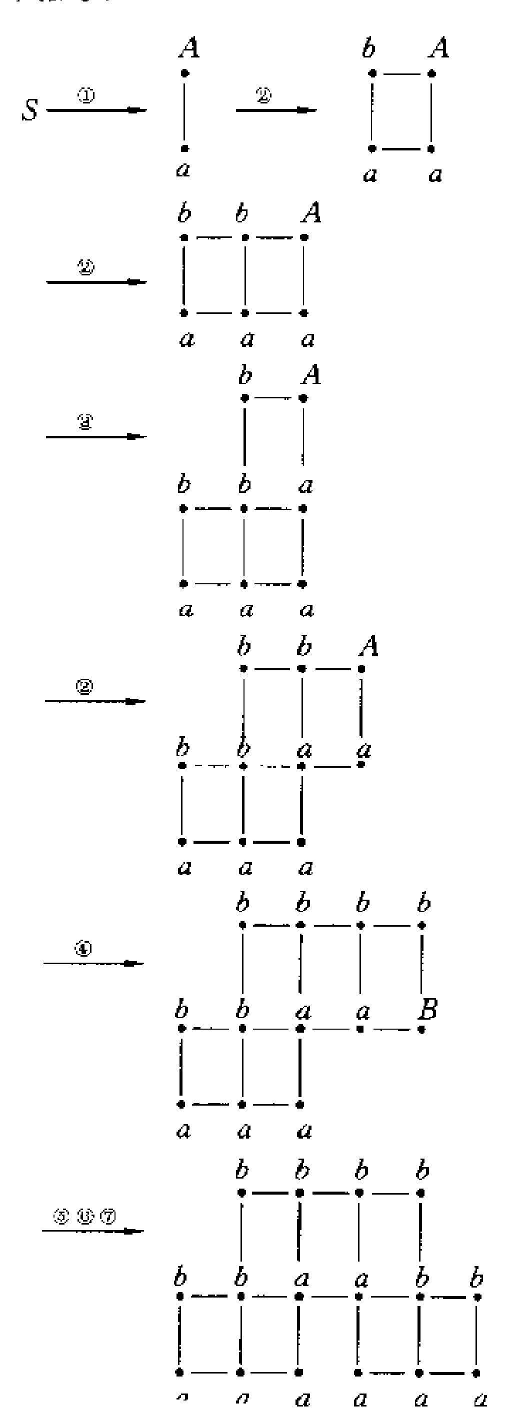


图 8-29　图文法推导结果

例 2



根据上述规则可做出如下推导：



在上述推导中，结构的上边缘都用标号 b，下边缘都用标号 a。总的描述规则由 (α,β,ϕ) 来规定，也就是用 β 代替 α，节点的连接则遵照 ϕ 的规定，这就是网文法的基本重写规则。

8.2.3 相似性描绘

图像描绘的另外一种途径可借助于与已知描绘子的相似程度来进行，这种方法可以在任何复杂的程度上建立相应的相似性测度。它可以比较两个简单的像素，也可以比较两个或两个以上的景物。

1. 距离测度

前面研究过的一些方法可以用来做为两幅图像区域之间进行比较的准则。例如，以矩做为

描绘子，假如两个区域的矩分别为 $\boldsymbol{X}_1$ 和 $\boldsymbol{X}_2$。把它们写成向量式如下：

$$\boldsymbol{X}_1 = \{x_1, x_2, \cdots, x_n\}$$
$$\boldsymbol{X}_2 = \{x_1', x_2', \cdots, x_n'\} \tag{8-70}$$

此时，$\boldsymbol{X}_1$ 和 $\boldsymbol{X}_2$ 之间的距离可定义如下：

$$D(\boldsymbol{X}_1, \boldsymbol{X}_2) = |\boldsymbol{X}_1 - \boldsymbol{X}_2|$$
$$= \sqrt{(\boldsymbol{X}_1 - \boldsymbol{X}_2)^{\mathrm{T}}(\boldsymbol{X}_1 - \boldsymbol{X}_2)} \tag{8-71}$$

采用距离这一测度可以测量两个描绘子之间的相似性。如果已知描绘子用 $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_L$ 表示，未知描绘子用 $\boldsymbol{X}$ 表示，可以计算 $\boldsymbol{X}$ 与已知描绘子的距离 $D(\boldsymbol{X}, \boldsymbol{X}_i)$，如果

$$D(\boldsymbol{X}, \boldsymbol{X}_i) < D(\boldsymbol{X}, \boldsymbol{X}_j) \tag{8-72}$$

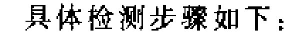
就可以判定 $\boldsymbol{X}$ 更接近第 i 个描绘子。式中 $j=1,2,\cdots,L$，并且。这个方法原则上可用于各种描绘子，只要它们能够用一矢量来表示就可以。

2. 相关性

当给定一幅大小为 $M \times N$ 的数字图像 $f(x,y)$，要确定它是否包含一个区域，该区域与某个大小为 $J \times K$ 中的某个区域 $w(x,y)$ 相类似，其中 $J<M$，$K<N$。解决这样问题常用的方法之一是求 $w(x,y)$ 和 $f(x,y)$ 之间的相关性。两个函数之间的相关的定义由下式表示：

$$R(m,n) = \sum_x \sum_y f(x,y) w(x-m, y-n) \tag{8-73}$$

其中 $m=0,1,2,\cdots,M-1$，$n=1,2,\cdots,N-1$。



具体检测步骤如下：

对于 $f(x,y)$ 中的任意值 (m,n) 用式(8-73)可求得一个 R 值，在 m,n 变化时，$w(x,y)$ 沿着图像移动，这时可得到 $R(m,n)$。求出 $R(m,n)$ 的最大值就说明 $w(x,y)$ 和 $f(x,y)$ 在此处最相似。但是在 m,n 接近边缘时，其精度较差。这个误差量正比于 $w(x,y)$ 的大小。上述步骤可由图 8-30 加以形象地说明。

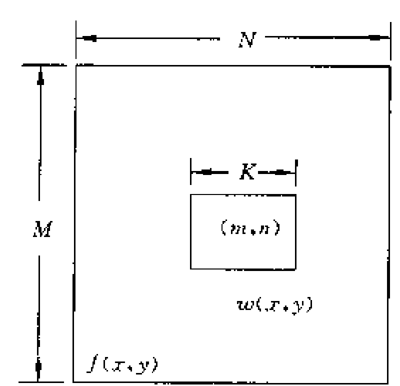


图 8-30 在给定点 (m,n) 上求 $f(x,y)$ 和 $w(x,y)$ 的相关性步骤

这里提到的相关检测法与前述的样板匹配法颇为相似。在这个意义下，样板就是 $w(x,y)$。相关检测法与样板匹配法的主要区别是 $w(x,y)$ 一般是一幅子图像。适合于图像特性的更复杂的相关定义可由下式表示：

$$R(m,n) = \frac{\sum_x \sum_y f(x,y) w(x-m, y-n)}{\left[\sum_x \sum_y f^2(x,y)\right]^{\frac{1}{2}}} \tag{8-74}$$

式中引入了归一化因子。归一化因子的计算是在 $w(x,y)$ 被划定的整个面积上进行的，因此它是作为位移函数而变化的。

相关性的计算可通过 FFT 算法在频域进行，这样比直接在空域计算更有效。

3. 结构相似性

一般来讲，结构相似性的描绘比起距离测度与相关性更难于公式化，因此，应用起来也就具有更高的难度。可以用作相似测度的典型的结构描绘子是线段的长度、线段之间的角度、亮度特性、区域的面积、在一幅图像中一个区域相对于另外一个区域的位置等等。例如，对一幅图

像可用出现于图像中的物体以及这些物体间的关系来描述图像，用于描述的一部分性质可能是下列的一种或几种：

1）亮度：黑、灰、白、亮、暗、均匀、有阴影等；

2）颜色：红、橙、黄、绿、青等；

3）结构：平滑、粒状、斑驳的、有条纹的等；

4）大小：长度、面积、体积、高度、宽度、深度、大小、高、矮、宽、窄等；

5）取向：水平、垂直、倾斜；

6）形状：实心的、中空的、密集的、参差不齐的、伸长的等。

上述的几种只反映了一个侧面，但是，就是这些，如果要从一幅图像中将它们抽取出来也是相当困难的。

基于结构分量之间关系的相似性测度，可以用某些文法将其公式化。假定有两类物体，可以分别用两种文法 G_1 和 G_2 来产生它们。给定一个特定的物体，如果它能用 G_1 来产生而不能用 G_2 来产生，那么就可以说这个物体更接近第一类。如果用两种文法都能产生，就不能分辨给定的物体了。所以，这种方法可以把文法近似于给定结构的紧密程度作为相似性的测度。

8.2.4 霍夫变换

霍夫(Hough)变换是一种线描述方法。它可以将笛卡儿坐标空间的线变换为极坐标空间中的点。图 8-31 是 x,y 坐标系中的一条直线。如果用 ρ 代表直线距原点的法线距离，θ 为该法线与 x 轴的夹角，则可用如下参数方程来表示该直线。这一直线的霍夫变换：

$$\rho = x\cos\theta + y\sin\theta \tag{8-75}$$

在极坐标域中便是如图 8-31(b)所示的一个点。由图 8-31(c)、(d)、(e)、(f)所示，在(x,y)坐标系中通过公共点的一簇直线，映射到(ρ,θ)坐标系中便是一个点集。在(x,y)坐标系中共线的点映射到(ρ,θ)坐标系便成为共点的一簇曲线。由此可见，霍夫变换使不同坐标系中的线和点建立了一种对应关系。

综上所述，可总结霍夫变换的几点性质如下：

1）(x,y)域中的一点对应于变换域(ρ,θ)中的一条正弦曲线。

2）变换域中的一点对应于(x,y)域中的一条直线。

3）(x,y)域中一条直线上的 n 个点对应于变换域中经过一个公共点的 n 条曲线。这条性质可证明如下。

证明 设(x,y)平面中的 n 个点(x_1,y_1)，(x_2,y_2)，…，(x_n,y_n)共一条直线，则有

$$y_i = ax_i + b \qquad i = 1,2,\cdots,n$$

由霍夫变换的定义可知，变换域的曲线为

$$\rho_i = x_i\cos\theta_i + y_i\sin\theta_i$$

将 $y_i = ax_i + b$ 代入上式，有：

$$\begin{aligned}\rho_i &= x_i\cos\theta_i + y_i\sin\theta_i \\ &= x_i\cos\theta_i + (ax_i + b)\sin\theta_i \\ &= x_i(\cos\theta_i + a\sin\theta_i) + b\sin\theta_i\end{aligned}$$

7

由此可知，无论 x_i 为何值，曲线都将通过 $\cos\theta_i + a\sin\theta_i = 0$这点，也就是$\left\{\theta = -\arctan\frac{1}{a}, \rho_i = b\sin\left(-\arctan\frac{1}{a}\right)\right\}$这一点。

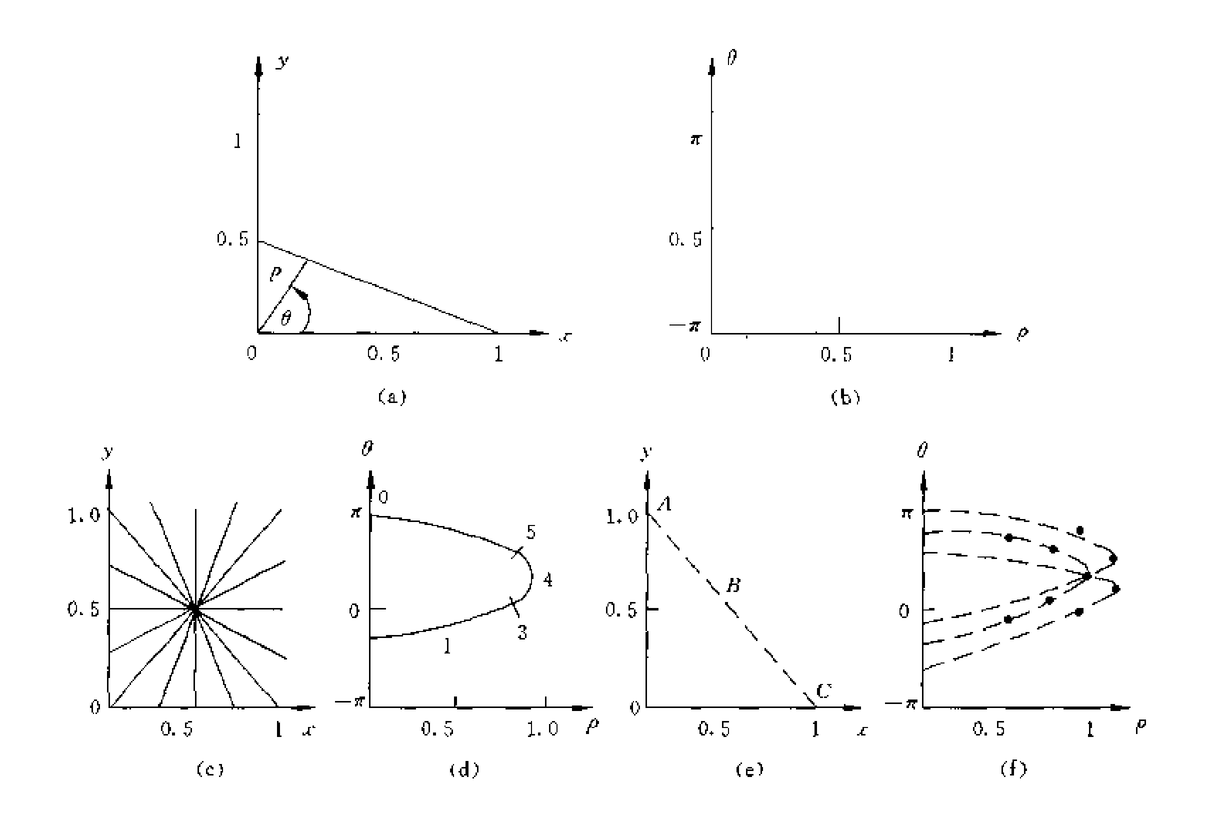


图 8-31 霍夫变换的原理

4）变换域中一条曲线上的 n 点对应于 (x,y) 域中过一公共点的 n 条直线。这条性质可证明如下。

证明 假设变换域中有 n 点 (ρ_1,θ_1)，(ρ_2,θ_2)，…，(ρ_n,θ_n) 在同一曲线上，则有：

$$\rho_i = a\cos\theta_i + b\sin\theta_i \qquad i=1,2,\cdots,n$$

对应于 (x,y) 域的直线可导出如下：

因为 $$\rho_i = a\cos\theta_i + b\sin\theta_i \qquad i=1,2,\cdots,n$$

所以，

$$\begin{aligned} y_i &= -x_i\cot\theta_i + \frac{1}{\sin\theta_i}\cdot\rho \\ &= -x_i\cot\theta_i + \frac{1}{\sin\theta_i}(a\cos\theta_i + b\sin\theta_i) \\ &= (-x_i + a)\cot\theta_i + b \end{aligned}$$

由此可见，不论 θ_i 为何值，直线都经过 $\{x_i=a, y_i=b\}$ 这一点。

霍夫变换的应用可用如下方法实现：

在 (x,y) 域中的每一离散数据点变换为 (ρ,θ) 域中的曲线。将 θ 和 ρ 分成许多小段，每一个 θ 小段和每一 ρ 小段构成一个小单元 $(\Delta\rho,\Delta\theta)$。对应于每一个小单元可设一累加器。在 (x,y) 域中可能落在直线上的每一点对应变换域中的一条曲线 $\rho = x_i\cos\theta + y_i\sin\theta$。分别使 θ 等于 $0,\Delta\theta$，$2\Delta\theta,3\Delta\theta,\cdots$，便可求出相应的 ρ 值，并分别计算落在各小单元中的次数，待全部 (x,y) 域内数据点变换完后，可对小单元进行检测，这样，落入次数较多的单元，说明此点为较多曲线的公共点，而这些曲线对应的 (x,y) 平面上的点可以认为是共线的。检测出 (x,y) 平面上 n 点后，将曲线交点坐标 (ρ_0,θ_0) 代入 $\rho_0 = x\cos\theta_0 + y\sin\theta_0$，便可得到逼近 n 点的直线方程。

在这种实现中，变换域小单元 $(\Delta\rho,\Delta\theta)$ 的大小直接影响 (x,y) 域中逼近直线的精度。霍夫变换的另外一个实用弱点是未考虑点的相邻性，有时得到的最佳逼近直线可能会由于邻近的

点的影响而产生扭曲。

作为霍夫变换的推广，可看到如下一些结果。例如，有一曲线方程为

$$Ax^2 + By^2 = C \tag{8-76}$$

显然，在椭圆上的每一点都满足式(8-76)。在此式中 x，y 是变量，A、B、C 是系数。如果把式(8-76)写成式(8-77)的形式，即

$$x^2A + y^2B = C \tag{8-77}$$

这里，把 A，B，C 看成变量，把 x^2，y^2看成系数，那么，在(x，y)域中的任何一点将对应于变换域中的一个曲面。(x，y)域中椭圆上的 n 点将对应于变换域中 n 个有共同交点的 n 个曲面。这一推广可用于圆的检测。

8.3 纹理分析

对纹理图像很难下一个确切的定义。类似于布纹、草地、砖砌地面等重复性结构的图像称为纹理图像。一般来说，纹理图像中灰度分布具有某种周期性，即便灰度变化是随机的，它也具有一定的统计特性。霍金斯认为纹理的标志有三要素：一是某种局部的序列性，在该序列更大的区域内不断重复；二是序列是由基本部分非随机排列组成的；三是各部分大致都是均匀的统一体，纹理区域内任何地方都有大致相同的结构尺寸。当然，这些也只是从感觉上看来是合理的，并不能得出定量的纹理测度。正因如此，对纹理特征的研究方法也是多种多样的，所提出的纹理特征参数效果如何也有待于进一步探讨。

8.3.1 纹理特征

纹理图像在很大范围内没有重大细节变化，在这些区域内图像往往显示出重复性结构。纹理可分为人工纹理和天然纹理。人工纹理是由自然背景上的符号排列组成，这些符号可以是线条、点、字母、数字等。自然纹理是具有重复排列现象的自然景像，如砖墙、种子、森林、草地之类的照片。人工纹理往往是有规则的，而自然纹理往往是无规则的。自然纹理和人工纹理的例子如图8-32所示，(a)是人工纹理图例，(b)是自然纹理图例。归纳起来，对纹理有两种看法，一是凭人们的直观印象，一是凭图像本身的结构。从直观印象出发包含有心理学因素，这样就会产生多种不同的统计纹理特性。从这一观点出发，纹理分析应该采用统计方法。从图像结构观点出发，则认为纹理是结构，根据这一观点，纹理分析应该采用句法结构方法。

描述图像特性的参数有很多种，对于纹理图像来说有必要知道各个像素及其邻近像素的灰度分布情况。了解邻近像素灰度值变化情况的最简单方法是取一阶微分、二阶微分值的平均值与方差，如果要考虑纹理的方向性特征，则可考查 θ 方向与 $\theta+\frac{\pi}{2}$ 方向差分的平均值与方差。

另外一种方法是检查小区域内的灰度直方图。例如，取小区域为 $n\times n$($n=3\sim7$)作这 n^2个像素的灰度直方图。然后检查各个小区域直方图的相似性。具有相似直方图的小区域同属于某一个大区域，而直方图不同的小区域分属不同的区域。检测灰度直方图相似性可以采用克尔玛哥诺夫-斯米诺夫检测法，这种方法先求两个灰度直方图的分布函数 $F_1(x)$和 $F_2(x)$，对所有的灰度值求分布函数的差，从中找出最大差值$\max\limits_{x}\{|F_2(x)-F_1(x)|\}$。若最大差值小于某一门限值，则认为这两个灰度直方图是相似的，具有相同的灰度分布，否则认为是不同的灰度直

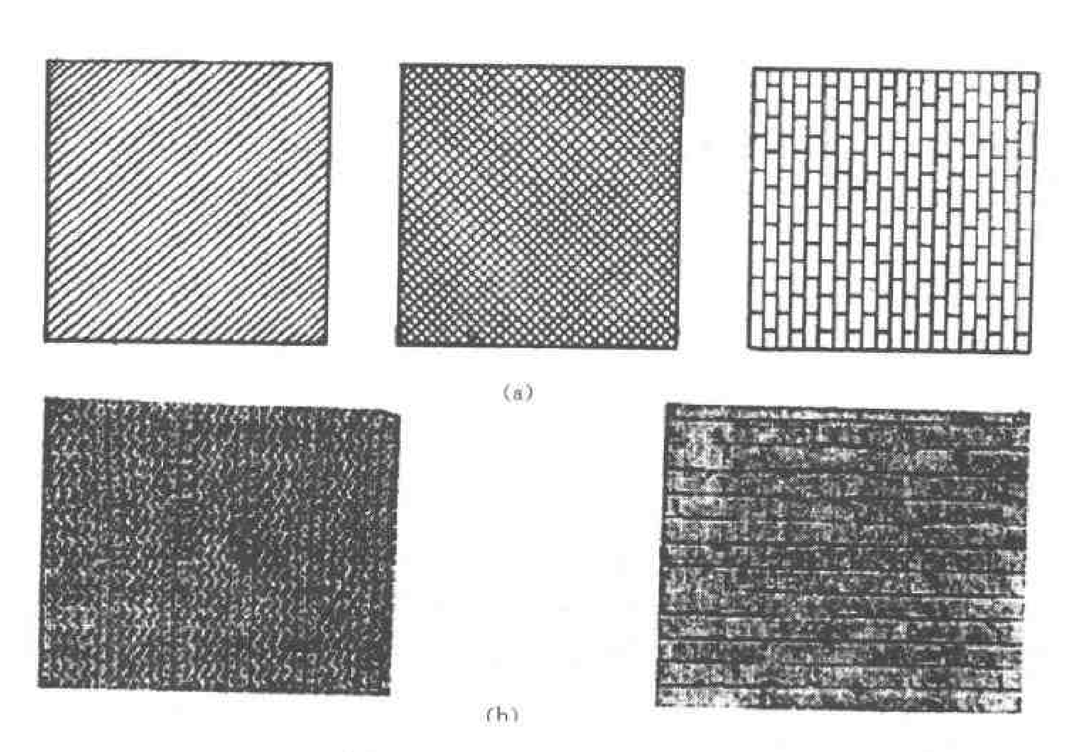


图8-32　人工纹理与自然纹理

方图。

8.3.2　用空间自相关函数作纹理测度

纹理常用它的粗糙性来描述。例如，在相同的观看条件下毛料织物要比丝织品粗糙。粗糙性的大小与局部结构的空间重复周期有关，周期大的纹理粗，周期小的纹理细。这种感觉上的粗糙与否不足以作为定量的纹理测度，但至少可以用来说明纹理测度变化的倾向。即小数值的纹理测度表示细纹理，大数值测度表示粗纹理。

用空间自相关函数作为纹理测度的方法如下：设图像为 $f(m,n)$，自相关函数可定义于下式：

$$C(\varepsilon,\eta,j,k)=\frac{\sum\limits_{m=j-w}^{j+w}\sum\limits_{n=k-w}^{k+w}f(m,n)f(m-\varepsilon,n-\eta)}{\sum\limits_{m=j-w}^{j+w}\sum\limits_{n=k-w}^{k+w}[f(m,n)]^2} \tag{8-78}$$

它是对 $(2w+1)\times(2w+1)$ 窗口内的每一像点 (j,k) 与偏离值为 $\varepsilon,\eta=0,\pm1,\pm2,\cdots,\pm T$ 的像素之间的相关值作计算。一般粗纹理区对给定偏离 (ε,η) 时的相关性要比细纹理区高，因而纹理粗糙性应与自相关函数的扩展成正比。自相关函数的扩展的一种测度是二阶矩，即：

$$T(j,k)=\sum_{\varepsilon=-T}^{j}\sum_{\eta=-T}^{k}\varepsilon^2\eta^2C(\varepsilon,\eta,j,k) \tag{8-79}$$

纹理粗糙性越大则 T 就越大，因此，可以方便地用 T 作为度量粗糙性的一种参数。

8.3.3　傅里叶功率谱法

计算纹理要选择窗口，仅一个点是无纹理可言的，所以纹理是二维的。设纹理图像为 $f(x,y)$，其傅里叶变换可由下式表示：

$$F(u,v)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(x,y)\exp\{-\mathrm{j}2\pi(ux+vy)\}\mathrm{d}x\mathrm{d}y \tag{8-80}$$

二维傅里叶变换的功率谱的定义如式(8-81)所示：

$$|F|^2=FF^* \tag{8-81}$$

式中 F^* 为 F 的共轭。功率谱 $|F|^2$ 反映了整个图像的性质。如果把傅里叶变换用极坐标形式表示，则有 $F(r,\theta)$ 的形式。如图8-33(a)所示，考虑到距原点为 r 的圆上的能量为

$$\Phi_r=\int_0^{2\pi}[F(r,\theta)]^2\mathrm{d}\theta \tag{8-82}$$

由此，可得到能量随半径 r 的变化曲线如图8-33(b)所示。对实际纹理图像的研究表明，在纹理较粗的情况下，能量多集中在离原点近的范围内，如图中曲线 A 那样，而在纹理较细的情况下，能量分散在离原点较远的范围内，如图中曲线 B 所示。由此可总结出如下分析规律：如果 r 较小，Φ_r 很大；r 很大时，Φ_r 反而较小，则说明纹理是粗糙的；反之，如果 r 变化对 Φ_r 的影响不是很大时，则说明纹理是比较细的。

另外，如图8-33(a)所示，研究某个 θ 角方向上的小扇形区域内的能量。这个能量随角度变化的规律可由下式求出：

$$\Phi_\theta=\int_0^{+\infty}|F(r,\theta)|^2\mathrm{d}r \tag{8-83}$$

当某一纹理图像沿 θ 方向的线、边缘等大量存在时，则在频率域内沿 $\theta+\frac{\pi}{2}$，即与 θ 角方向成直角的方向上能量集中出现。如果纹理不表现出方向性，则功率谱也不呈现方向性。因此 $|F|^2$ 值可以反映纹理的方向性。

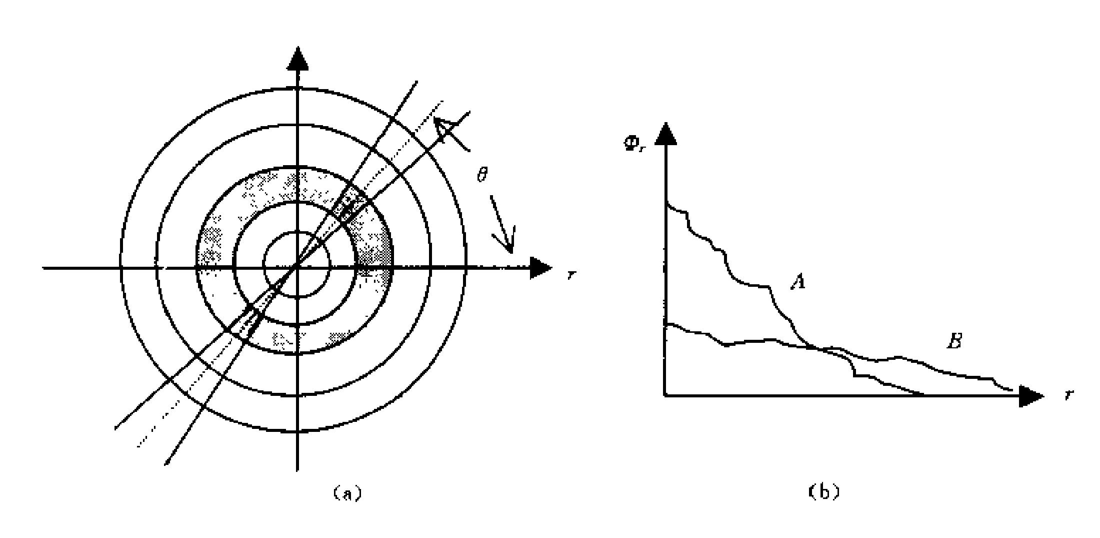


图8-33　纹理图像的功率谱分析

8.3.4　联合概率矩阵法

联合概率矩阵法是对图像所有像素进行统计调查，以便描述其灰度分布的一种方法。

取图像中任意一点 (x,y) 及偏离它的另一点 $(x+a,y+b)$，设该点对的灰度值为 (g_1,g_2)。令点 (x,y) 在整个画面上移动，则会得到各种 (g_1,g_2) 值，设灰度值的级数为 k，则 g_1 与 g_2 的组合共有 k^2 种。对于整个画面，统计出每一种 (g_1,g_2) 值出现的次数，然后排列成一个方阵，再用 (g_1,g_2) 出现的总次数将它们归一化为出现的概率 $p(g_1,g_2)$，称这样的方阵为联合概率矩阵。

图8-34 为一个示意的简单例子。图8-34 (a)为原图像，灰度级为16级，为使联合概率矩

阵简单些，首先将灰度级数减为4级。这样，图8-34（a）变为图8-34（b）的形式。g_1、g_2分别取值为0，1，2，3，由此，将(g_1,g_2)各种组合出现的次数排列起来，就可得到图8-34(c)～(e)所示的联合概率矩阵。

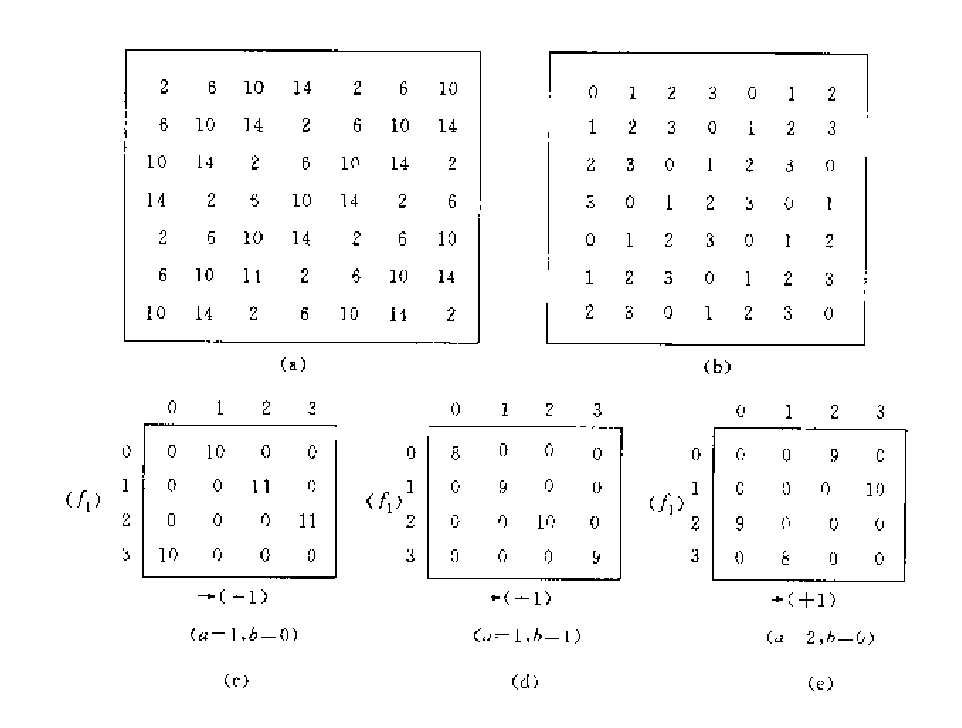


图8-34 联合概率矩阵计算示例

由此可见，距离差分值(a,b)取不同的数值组合，可以得到不同的情况下的联合概率矩阵。a,b的取值要根据纹理周期分布的特性来选择，对于较细的纹理，选取(1，0)，(1，1)，(2，0)等这样小的差分值是有必要的。当a,b取值较小时，对应于变化缓慢的纹理图像，其联合概率矩阵对角线上的数值较大，而纹理的变化越快，则对角线上的数值越小，而对角线两侧上的元素值增大。为了能描述纹理的状况，有必要选取能综合表现联合概率矩阵状况的参数，典型的有以下几种：

$$Q_1 = \sum_{g_1}\sum_{g_2}[p(g_1,g_2)]^2 \tag{8-84}$$

$$Q_2 = \sum_{k}k^2\left[\sum_{g_1}\sum_{g_2}p(g_1,g_2)\right] \tag{8-85}$$

$$k = |g_1 - g_2|$$

$$Q_3 = \frac{\sum_{g_1}\sum_{g_2}g_1g_2p(g_1,g_2) - \mu_x\mu_y}{\sigma_x\sigma_y} \tag{8-86}$$

$$Q_4 = -\sum_{g_1}\sum_{g_2}p(g_1,g_2)\lg p(g_1,g_2) \tag{8-87}$$

其中

$$\mu_x = \sum_{g_1}g_1\sum_{g_2}p(g_1,g_2) \tag{8-88}$$

$$\mu_y = \sum_{g_2} g_2 \sum_{g_1} p(g_1, g_2) \tag{8-89}$$

$$\sigma_x^2 = \sum_{g_1} (g_1 - \mu_x)^2 \sum_{g_2} p(g_1, g_2) \tag{8-90}$$

$$\sigma_y^2 = \sum_{g_2} (g_2 - \mu_y)^2 \sum_{g_1} p(g_1, g_2) \tag{8-91}$$

这些参数到底都代表着哪一种图像特性并不是直观的，但这些参数用来描述纹理特性还是相当有效的。

8.3.5 灰度差分统计法

设(x,y)为图像中的一点，该点与和它只有微小距离的点$(x+\Delta x, y+\Delta y)$的灰度差值为

$$g_\Delta(x,y) = g(x,y) - g(x+\Delta x, y+\Delta y) \tag{8-92}$$

g_Δ 称为灰度差分。设灰度差分值的所有可能取值共有 m 级，令点(x,y)在整个画面上移动，计出 $g_\Delta(x,y)$取各个数值的次数，由此可以做出 $g_\Delta(x,y)$的直方图。由直方图可以知道 $g_\Delta(x,y)$取值的概率 $p_\Delta(i)$。

当取较小 i 值的概率 $p_\Delta(i)$较大时，说明纹理较粗糙；概率较平坦时，说明纹理较细。

一般采用下列参数来描述纹理图像的特性：

(1) 对比度

$$\text{CON} = \sum_i i^2 p_\Delta(i) \tag{8-93}$$

(2) 角度方向二阶矩

$$\text{ASM} = \sum_i [p_\Delta(i)]^2 \tag{8-94}$$

(3) 熵

$$\text{ENT} = -\sum_i p_\Delta(i) \lg p_\Delta(i) \tag{8-95}$$

(4) 平均值

$$\text{MEAN} = \frac{1}{m} \sum_i i p_\Delta(i) \tag{8-96}$$

在上述各式中，$p_\Delta(i)$较平坦时，ASM 较小，ENT 较大，$p_\Delta(i)$越分布在原点附近，则 MEAN 值越小。

8.3.6 行程长度统计法

设点(x,y)的灰度值为 g，与其相邻的点的灰度值可能也为 g。统计出从任一点出发沿 θ 方向上连续 n 个点都具有灰度值 g 这种情况发生的概率，记为 $p(g,n)$。在某一方向上具有相同灰度值的像素个数称为行程长度(run length)。由 $p(g,n)$可以引出一些能够较好地描述纹理图像变化特性的参数。

(1) 长行程加重法

$$\text{LRE} = \frac{\sum_{g,n} n^2 p(g,n)}{\sum_{g,n} p(g,n)} \tag{8-97}$$

(2) 灰度值分布

$$\mathrm{GLD}=\frac{\sum_{g}\left[\sum_{n}p(g,n)\right]^{2}}{\sum_{g,n}p(g,n)} \tag{8-98}$$

(3) 行程长度分布

$$\mathrm{RLD}=\frac{\sum_{g}\left[\sum_{n}p(g,n)\right]}{\sum_{g,n}p(g,n)} \tag{8-99}$$

(4) 行程比

$$\mathrm{RPG}=\frac{\sum_{g,n}p(g,n)}{N^{2}} \tag{8-100}$$

其中 N^2为像素总数。

8.3.7 其他几种方法

除前述的几种纹理统计分析法外，还有一些其他方法。如线性预测系数法。这种方法是将某点(i,j)的灰度 g_{ij}由相邻接的8个点的灰度值来预测，g_{ij}的预测值为 $\hat{g}_{ij}$，

$$\begin{aligned}\hat{g}_{ij}= & a_1g_{i-1,j-1}+a_2g_{i-1,j}+a_3g_{i-1,j+1}+a_4g_{i,j-1}\\ & +a_5g_{i,j+1}+a_6g_{i+1,j-1}+a_7g_{i+1,j}+a_8g_{i-1,j-1}\end{aligned} \tag{8-101}$$

可以选择系数 $a_1,a_2,\cdots,a_8$，使 g_{ij}值与预测值 $\hat{g}_{ij}$的差为最小。对于不同的纹理可得到不同的系数矢量 $\boldsymbol{A}$，$\boldsymbol{A}=\{a_1,a_2,\cdots,a_8\}$。由此，可用不同矢量间的距离来区别不同的纹理。

还有一种像素点组合的方法。这种方法是先确定检查纹理特性所需要的像素组合，这种组合出现的灰度模式为 X，然后，对各种纹理模式 $w_1,w_2,\cdots,w_n$，去检查这种像素组合灰度模式 X 出现的概率 $P(X/w_1)$、$P(X/w_2)$，$\cdots$，$P(X/w_n)$。从这些概率中找出最大者，则可以认为所检查的纹理图像的纹理属于概率值最大的那一类纹理模式。

8.3.8 纹理的句法结构分析法

在纹理的句法结构分析中把纹理定义为结构基元按某种规则重复分布所构成的模式。为了分析纹理结构，首先要描述结构基元的分布规则。一般可做如下两项工作：(1)从输入图像中提取结构基元并描述其特征，(2)描述结构基元的分布规则。具体做法是首先把一张纹理图片分成许多窗口，也就是形成子纹理。最小的小块就是最基本的子纹理，即基元。纹理基元可以是一个像素，也可以是4个或9个灰度比较一致的像素集合。纹理的表达可以是多层次的，如图8-35(a)所示。它可以从像素或小块纹理一层一层地向上拼合。当然，基原的排列可有不同的规则，如图8-35(b)所示，第一级纹理排列为甲乙甲，第二级排列为乙甲乙等等，其中甲、乙代表基原或子纹理，这样就组成一个多层的树状结构，可用树状文法产生一定的纹理并用句法加以描述，纹理的树状安排可有多种方法。第一种方法如图8-35(c)所示，树根安排在中间，树枝向两边伸出，每个树枝有一定的长度。当窗口中像点数为奇数时用这种排列比较方便，此时，每个分枝长度相同。第二种方法如图8-35(d)所示，树根安排在一侧，分枝都向另一侧伸展，这种排列对奇数像点和偶树数像点都适用。

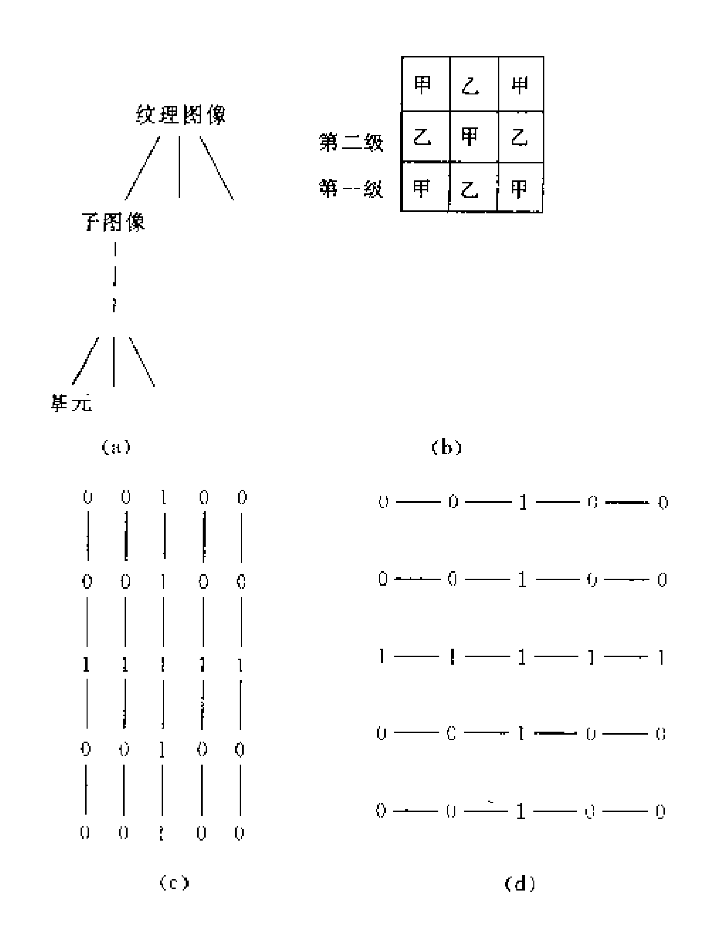


图8-35 纹理的树状描述及排列

纹理判别可用如下办法：

首先把纹理图像分成固定尺寸的窗口，用树状文法说明属于同纹理图像的窗口，识别树状结构可以用树状自动机，因此，对每一个纹理文法可建立一个"结构保存的误差修正树状自动机"。该自动机不仅可以接受每个纹理图像中的树，而且能用最小距离判据，辨识类似的有噪声的树。以后，可以对一个分割成窗口的输入图像进行分类。

8.4 形状分析的细线化

形状是图形分析中常遇到的概念。形状分析中的重要环节是细线化及骨骼化。

图形中的线条一般都有一定的宽度，在直接处理这样的图形会有一些麻烦。在提取线的分枝、弯曲等特征时，线的宽度也会带来不便。为此，有必要从这样的线条中找出位于中央部位的宽度大致为一个像素的中心线来。提取中心线的结果如图8-36所示。求中心线的方法之一是线条图形的细线化处理。细线化的基本方法是针对图形边缘上的点，首先观察其相邻点的状况，在不破坏图形连结性的情况下，逐渐消去位于边缘的这些点。例如，可以用图8-37所示的模板。这个模板可以旋转90°、180°、270°使用。如果在图形的边缘处有符合该模板的部分，则把中心点的1强制变为0。用这样的方法即可消去边缘的点，实现细线化。用这种方法进行细线化，一般要按照从上至下，从左至右，从下至上，从右至左的顺序对图形反复处理多次之后才能逐步得到宽度近似为一个像素的中心线。

这种细化方法要准备多种细化模板，操作较为麻烦。另外一种简单的细化方法如图8-38

```
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000001100000
000000000000000000000000000000011XX00000
00000000001111110000000000111111XX100000
0000000000111111100000000011111XX1100000
000000000111111110000000001111XX11000000
00000000011111111001111001111XX111000000
0000000111111111111111111111XX1110000000
000001111111111111111111111XX11110000000
00000111111111111111111111XX111100000000
00000111111XXXXXXXXXXXX111X1111000000000
0000011111XX1111111111XXXXX1111000000000
0000011111X111111111111111X1111000000000
000001111XX111111111111111XX111000000000
000001111X11111111101111111X111000000000
00000111XX11111100000111111XX11000000000
00000111X1110000000000001111111100000000
0000011XX1100000000000000011111100000000
0000011X11000000000000000001111100000000
0000011X11000000000000000001110000000000
0000011X11000000000000000000000000000000
0000011XX1110000000000000000000000000000
00000111X1111000000000110000000000000000
00000111XX111000011111111000000000000000
000001111X1111111111111X1000000000000000
000001111XX11111111111XX1100000000000000
0000011111XXXXXXXXXX1XX11100000000000000
000001111XX11111111XXX111100000000000000
000001111X1111111111XXX11110000000000000
000001111X1110000111X1XX1110000000000000
00000111XX111000011XX11X1110000000000000
000001111111000011XX111XX110000000000000
00000111111100011XX11111X110000000000000
00000111110000111XX1000111X1110000000000
0000000111000011X1100011111100000000000
0000000000000001X00000111111000000000000
0000000000000011X00000011111000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
```

图8-36 细化处理的例子(X 为中心线)

所示。第一步，将左边用L标志的边界像素去掉，只要它们不是孤点而且也不破坏八连通性的都可以去掉。第二步，如果与左边的条件一样，可用同样的方法把右边的用R标志的边缘像素去掉。然后，同样是只要不是孤点而且去除后不致破坏八连通性的顶部点（以T标志）和底部

点（以 B 标志）都去掉。这样对边缘点进行反复处理即可实现细线化。

无论用哪种细化法都会存在一些困难问题，即边缘存在有毛刺或突起时，细化的结果会在这些突起处形成分枝线。

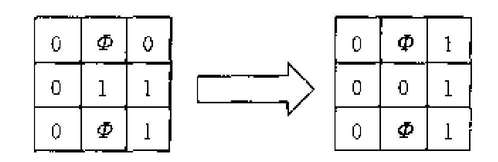


图8-37　细化处理模板（Φ 表示0或1）

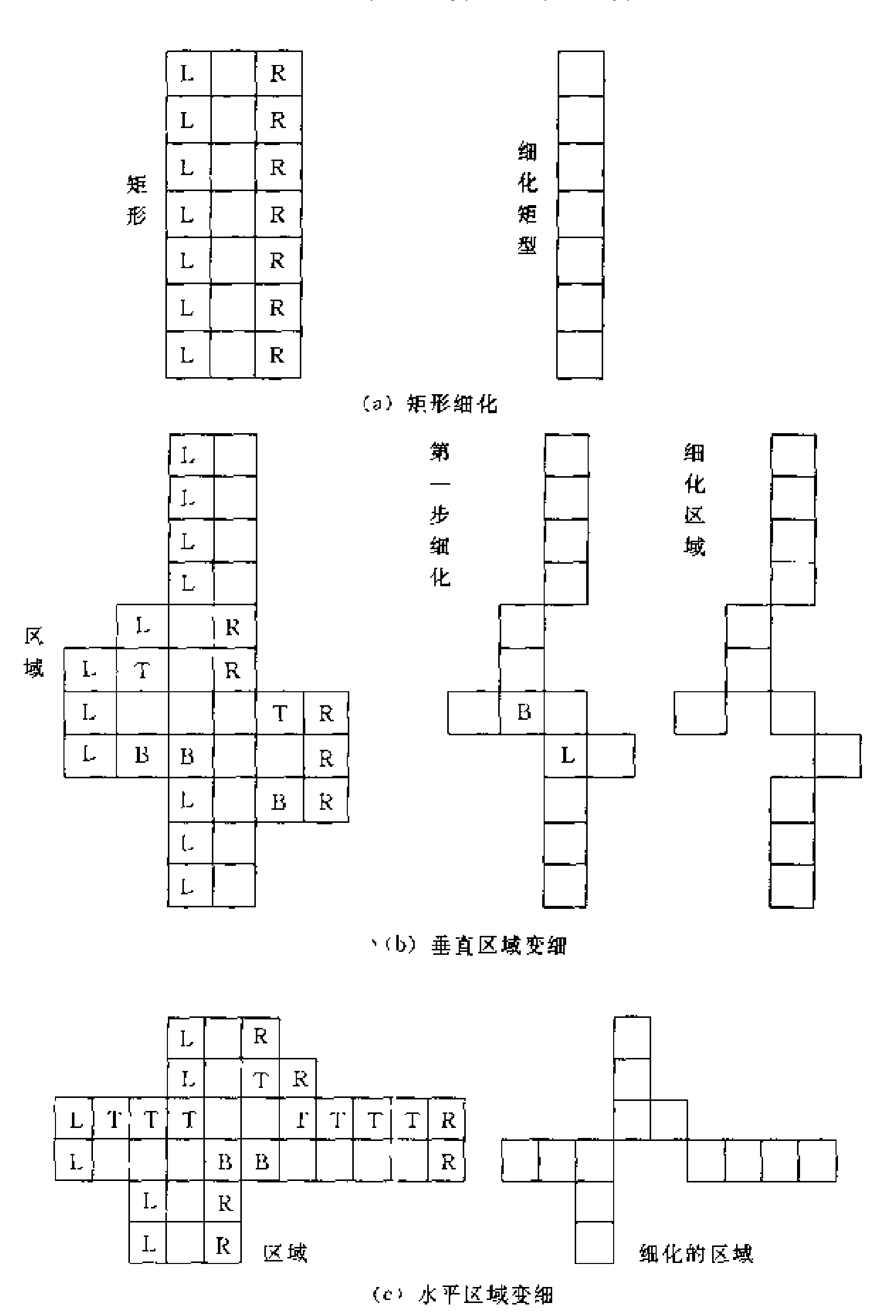


图8-38　细化算法

思 考 题

1. 试设计一个具有双峰直方图特性图像的最佳分割点的程序。

2. 如果图像背景和目标灰度分布均有正态分布特性，其均值分别为 μ 和 υ，而且图像与背景面积相等，证明最佳阈值点为$\frac{\mu+\upsilon}{2}$。

3. 如果图像灰度由于照明原因引起不均匀，如何求得最佳分割效果，试述几种可能的解决方案。

4. 如何设计检测线和点的模板，举例说明。

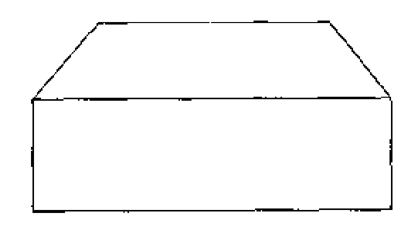
5. 试述区域生长分割算法的基本原理。

6. 对描绘子的基本要求是什么？有哪几种常用的描绘子？

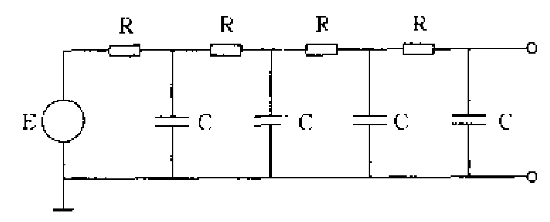
7. 什么是形式语言？它有什么用途？

8. 试述文法类型及其之间的关系。

9. 试用图片描绘语言描绘下图。



10. 试用树文法描绘下列电路。



11. 霍夫变换检测线的主要弱点是什么？如何克服之？

12. 纹理的定义是什么？如何描述纹理？

13. 傅里叶功率谱法描绘纹理的基本原理及其判断规律是什么？

14. 试述用自相关函数作纹理测度的基本原理。

15. 有如下之纹理图像，试求其联合概率矩阵。

$$\begin{matrix} 0 & 1 & 2 & 0 & 1 & 2 \\ 1 & 2 & 0 & 1 & 2 & 0 \\ 2 & 0 & 1 & 2 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 & 2 \\ 1 & 2 & 0 & 1 & 2 & 0 \\ 2 & 0 & 1 & 2 & 0 & 1 \end{matrix}$$

第9章 数学形态学原理

9.1 数学形态学的发展

数学形态学(Mathematical Morphology)是一种应用于图像处理和模式识别领域的新的方法。形态学是生物学的一个分支,常用它来处理动物和植物的形状和结构。数学形态学的历史可追溯到19世纪的Eular、Steiner、Crofton和本世纪的Minkowski。1964年,法国学者Serra对铁矿石的岩相进行了定量分析,以预测铁矿石的可轧性。几乎在同时,Matheron研究了多孔介质的几何结构、渗透性及两者的关系,他们的研究成果直接导致数学形态学雏形的形成。随后、Serra和Matheron在法国共同建立了枫丹白露(Fontainebleau)数学形态学研究中心。在以后几年的研究中,他们逐步建立并进一步完善了数学形态学的理论体系,此后,又研究了基于数学形态学的图像处理系统。

数学形态学是一门建立在严格的数学理论基础上的科学。Matheron于1973年出版的《Ensembles Aleatoireset Geometrie Integrate》一书中严谨而详尽地论证了随机集论和积分几何,为数学形态学奠定了理论基础。1982年,Serra出版的专著《Image Analysis and Mathematical Morphology》是数学形态学发展的里程碑,它表明数学形态学在理论上已趋于完备,在实际应用中不断深入。此后,经过科学工作者的不断努力,Serra主编的《Image Analysis and Mathematical Morphology》第2、3卷相继出版。1986年,CVGIP(Computer Vision Graphics and Image Processing)发表了数学形态学专辑,从而使得数学形态学的研究呈现了新的景象。同时,枫丹白露研究中心的学者们又相继提出了基于数学形态学方法的纹理分析模型系列,从而使数学形态学的研究前景更加光明。

随着数学形态学逻辑基础的发展,其应用开始向边缘学科和工业技术方面发展。数学形态学的应用领域已不限于传统的微生物学和材料学领域,80年代初又出现了几种新的应用领域,如:工业控制、放射医学、运动场景分析等。数学形态学在我国的应用研究也很快,目前,已研制出一些以数学形态学为基础的实用图像处理系统,如:中国科学院生物物理研究所和计算机技术研究所负责,由软件研究所、电子学研究所和自动化研究所参加研究的癌细胞自动识别系统等。

数学形态学是一门综合了多学科知识的交叉科学,其理论基础颇为艰深,但其基本观念却比较简单。它体现了逻辑推理与数学演绎的严谨性,又要求具备与实践密切相关的实验技术与计算技术。它涉及微分几何、积分几何、测度论、泛函分析和随机过程等许多数学理论,其中积分几何和随机集论是其赖以生存的基石。总之,数学形态学是建立在严格的数学理论基础上而又密切联系实际的科学。

用于描述数学形态学的语言是集合论,因此,它可以提供一个统一而强大的工具来处理图像处理中所遇到的问题。利用数学形态学对物体几何结构的分析过程就是主客体相互逼近的过程。利用数学形态学的几个基本概念和运算,将结构元灵活地组合、分解,应用形态变换序列达到分析的目的。

利用数学形态学进行图像分析的基本步骤有如下几步：

1）提出所要描述的物体几何结构模式，即提取物体的几何结构特征；

2）根据该模式选择相应的结构元素，结构元素应该简单而对模式具有最强的表现力；

3）用选定的结构元对图像进行击中与否（HMT）变换，便可得到比原始图像显著突出物体特征信息的图像。如果赋予相应的变量，则可得到该结构模式的定量描述；

4）经过形态变换后的图像突出我们需要的信息，此时，就可以方便地提取信息。

数学形态学方法比其他空域或频域图像处理和分析方法具有一些明显的优势。如：在图像恢复处理中，基于数学形态学的形态滤波器可借助于先验的几何特征信息，利用形态学算子有效地滤除噪声，又可以保留图像中的原有信息。另外，数学形态学算法易于用并行处理方法有效地实现，而且硬件实现容易；基于数学形态学的边缘信息提取处理优于基于微分运算的边缘提取算法，它不像微分算法对噪声那样敏感，同时，提取的边缘也比较光滑；利用数学形态学方法提取的图像骨架也比较连续，断点少。

数学形态学的核心运算是击中与否变换（HMT），在定义了 HMT 及其基本运算膨胀（Dilation）和腐蚀（Erosion）后，再从积分几何和体视学移植一些概念和理论，根据图像分析的各种要求，构造出统一的、相同的或变化很小的结构元素进行各种形态变换。在形态算法设计中，结构元的选择十分重要，其形状、尺寸的选择是能否有效地提取信息的关键。一般情况，结构元的选择本着如下几个原则进行：

1）结构元必须在几何上比原图像简单，且有界。当选择性质相同或相似的结构元时，以选择极限情况为益；

2）结构元的凸性非常重要，对非凸子集，由于连接两点的线段大部分位于集合的外面，故而用非凸子集作为结构元将得不到什么信息。

总之，数学形态学的基本思想和基本研究方法具有一些特殊性，掌握和运用好这些特性是取得良好结果的关键。

9.2 数学形态学的基本概念和运算

在数学意义上，我们用形态学来处理一些图像，用以描述某些区域的形状如边界曲线、骨架结构和凸形外壳等。另外，我们也用形态学技术来进行预测和快速处理如形态过滤、形态细化、形态修饰等。而这些处理都是基于一些基本运算实现的。

用于描述数学形态学的语言是集合论。数学形态学最初是建立在集合论基础上的代数系统，它提出了一套独特的变换和概念用于描述图像的基本特征。这些数学工具是建立在积分几何和随机集论的基础之上，这决定了它可以得到几何常数的测量和反映图像的体视性质。

集合代表图像中物体的形状，例如：在二进制图像中所有黑色像素点的集合就是对这幅图像的完整描述。在二进制图像中，当前集合指二维整形空间的成员，集合中的每个元素都是一个二维变量，用(x,y)表示，按规则代表图像中的一个黑色像素点。灰度数字图像可以用三维集合来表示。在这种情况下，集合中每个元素的前两个变量用来表示像素点的坐标，第三个变量代表离散的灰度值。在更高维数的空间集合中可以包括其他的图像属性，如颜色和时间。

形态运算的质量取决于所选取的结构元和形态变换。结构元的选择要根据具体情况来确定，而形态运算的选择必须满足一些基本约束条件。这些约束条件称为图像定量分析的原则。

9.2.1 数学形态学定量分析原则

1. 平移兼容性

设待分析图像为 X，Φ 表示某种图像变换或运算，$\Phi(X)$ 表示 X 经变换或运算后的新图像。设 $\boldsymbol{h}$ 为一矢量，X_h 表示将图像 X 平移一个位移矢量后的结果，那末，平移兼容性原则可表示为

$$\Phi(X_h) = [\Phi(X)]_h \tag{9-1}$$

此式说明图像 X 先平移然后变换的结果与图像先变换后平移的结果是一样的。

2. 尺度变换兼容性

设缩放因子 λ 是一个正的实常数，λX 表示对图像 X 所做的相似变换，则尺度变换兼容性原则可表示如下：

$$\lambda\Phi\left(\frac{1}{\lambda}X\right) = \Phi_\lambda(X) \tag{9-2}$$

如果设图像运算 Φ 为结构元 B 对 X 的腐蚀($X\ominus B$)，则 Φ_λ 为结构元 λB 对 X 的腐蚀，则上式可具体化为

$$\lambda\left(\frac{1}{\lambda}X \ominus B\right) = X \ominus \lambda B \tag{9-3}$$

3. 局部知识原理

如果 Z 是一个图形("闭集")，则相对于 Z 存在另一个闭集 Z'，使得对于图形 X 有下式成立：

$$(\Phi(X \cap Z)) \cap Z' = \Phi(X) \cap Z' \tag{9-4}$$

在物理上，可以将 Z 理解为一个"掩模"。在实际中，观察某一个对象时，每次只能观察一个局部，即某一掩模覆盖的部分 $X\cap Z$。该原则要求对每种确定的变换或运算 Φ，当掩模 Z 选定以后，都能找到一个相应的模板 Z'，使得通过 Z' 所观察到的局部性质，即$(\Phi(X\cap Z))\cap Z'$ 与整体性质 $\Phi(X)\cap Z'$ 相一致。

4. 半连续原理

在研究一幅图像时，常采用逐步逼近的方法，即对图像 X 的研究往往需要通过一系列图像 X_1、X_2、…、X_n、…的研究实现，其中诸个 X_n 逐步逼近 X。半连续原理要求各种图像变换后应满足这样的性质：对真实图像 X 的处理结果应包含在对一系列图像 X_n 的处理结果内。

5. 形态运算的基本性质

除了一些特殊情况外，数学形态学处理一般都是不可逆的。实际上，对图像进行重构的思想在该情况下是不恰当的。任何形态处理的目的都是通过变换去除不感兴趣的信息，保留感兴趣的信息。在形态运算中的几个关键性质如下：

递增性：
$$X \subset Y \to \Psi(X) \subset \Psi(Y) \qquad \forall X,Y \in \mathscr{P}(E) \tag{9-5}$$

反扩展性：
$$\Psi(X) \subset X \qquad \forall X \in \mathscr{P}(E) \tag{9-6}$$

幂等性：
$$\Psi[\Psi(X)] = \Psi(X) \qquad \forall X \in \mathscr{P}(E) \tag{9-7}$$

其中：Ψ 表示形态变换，$\mathscr{P}(E)$ 表示 Euclidean 空间 E 的幂集。

9.2.2 数学形态学的基本定义及基本算法

集合论是数学形态学的基础，在这里我们首先对集合论的一些基本概念作一总结性的概

括介绍。对于形态处理的讨论，我们将从两个最基本的模加处理和模减处理开始。它们是以后大多数形态处理的基础。

1. 基本定义

(1) 集合

具有某种性质的确定的有区别的事物的全体。如果某种事物不存在，称为空集。集合常用大写字母 $A,B,C,\cdots$ 表示，空集用 $\varnothing$ 表示。

设 E 为一自由空间，$\mathscr{P}(E)$ 是由集合空间 E 所构成的幂集，集合 $X,B\in\mathscr{P}(E)$，则集合 X 和 B 之间的关系只能有以下三种形式：

① 集合 B 包含于 X(表示为 $B\subset X$)；

② 集合 B 击中 X(表示为 $B\Uparrow X$)，即 $B\cap X\neq\varnothing$；

③ 集合 B 相离于 X(表示为 $B\subset X^c$)，即 $B\cap X=\varnothing$；

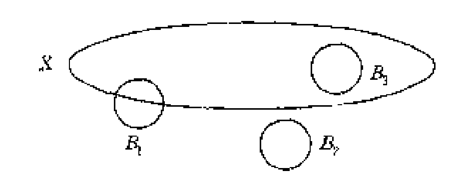


图 9-1 B_1 击中 X，B_2 相离于 X，B_3 包含于 X

(2) 元素

构成集合的每一个事物称之为元素。元素常用小写字母 $a,b,c,\cdots$ 表示，应注意的是任何事物都不是空集的元素。

(3) 平移转换

设 A 和 B 是两个二维集合，A 和 B 中的元素分别是

$$a=(a_1,a_2)\qquad b=(b_1,b_2)$$

定义 $x=(x_1,x_2)$，对集合 A 的平移转换为

$$(A)_x=\{c\mid c=a+x,\ \text{for } a\in A\}\tag{9-8}$$

(4) 子集

当且仅当集合 A 的所有元素都属于 B 时，称 A 为 B 的子集。

(5) 补集

定义集合 A 的补集为

$$A^c=\{x\mid x\notin A\}\tag{9-9}$$

(6) 差集

定义集合 A 和 B 的差集为

$$A-B$$

$$A-B=\{x\mid x\in A,x\notin B\}=A\cap B^c\tag{9-10}$$

(7) 映像

定义集合 B 的映像为 $\hat{B}$

$$\hat{B}=\{x\mid x=-b,b\in B\}\tag{9-11}$$

(8) 并集

由 A 和 B 的所有元素组成的集合称为 A 和 B 的并集。

(9) 交集

由 A 和 B 的公共元素组成的集合称为 A 和 B 的交集。

图 9-2 解释了刚才几个定义，图中的黑点为集合的原点。图 9-2(a)显示集合 A；图 9-2(b)表示 A 被 $x=(x_1,x_2)$平移，注意平移是在 A 的每个元素上加上 $x=(x_1,x_2)$。图 9-2(c)表示集合 B；图 9-2(d)显示了 B 关于原点的反转。图 9-2(e)显示了集合 A 及其补，最后图 9-2(f)显示了图 9-2(e)的集合 A 与图 9-2(f)中的集合 B 的差。

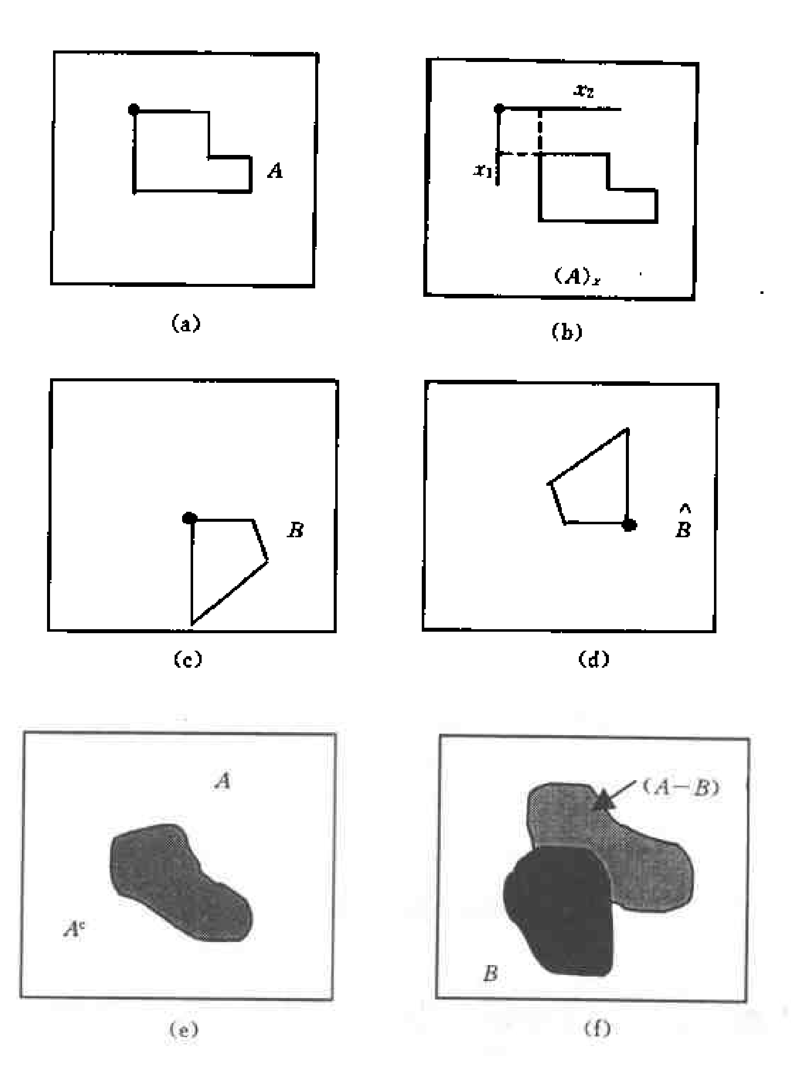


图 9-2 (a)集合 A，(b)用 x 平移集合 A 后的结果，(c)集合 B，(d)B 的反转，(e)集合 A 和它的补集，(f)两个集合的差集(如阴影所示)
前 4 幅图的黑点表示了每个集合的起点

2. 膨胀

A,B 为 Z^2 中的集合，$\varnothing$为空集，A 被 B 的膨胀，记为 $A\oplus B$，$\oplus$为膨胀算子，膨胀的定义为

$$A\oplus B=\{x\mid[(\hat{B})_x\cap A]\neq\varnothing\} \tag{9-12}$$

该式表明的膨胀过程是 B 首先做关于原点的映射，然后平移 x。A 被 B 的膨胀是 $\hat{B}$ 被所有 x 平移后与 A 至少有一个非零公共元素。根据这个解释，公式(9-12)可以重写如下：

$$A\oplus B=\{x\mid[(\hat{B})_x\cap A]\subseteq\varnothing\} \tag{9-13}$$

同在其他的形态处理中一样，集合 B 在膨胀操作中通常被称为结构元素。

公式(9-12)不是现在形态学文献中膨胀的惟一定义。然而，前面这个定义有一个明显的优

势，因为当结构元素 B 被看为卷积模板时有更加直观的概念。尽管膨胀是基于集合的运算，而卷积是基于算术运算，但是 B 关于原点的"映射"及而后连续的平移使它可以滑过集合(图像) A 的基本过程类似于卷积过程。

图 9-3(a)表示一个简单的集合，图 9-3(b)表示一个结构元素及其"映射"。在此图情况下，因为结构元素 B 关于原点对称，所以，结构元素 B 及其映射 $\hat{B}$ 相同。图 9-3(c)中的虚线表示作为参考的原始集合，实线示出若 $\hat{B}$ 的原点平移至 x 点超过此界限，则 $\hat{B}$ 与 A 的交集为空。这样实线内的所有点构成了 A 被 B 的膨胀。图 9-3(d)表示预先设计的一个结构元素，其目的是为了得到一个垂直膨胀比水平膨胀大的结果。图 9-3(e)显示为用此构成元素膨胀后得到的结果。

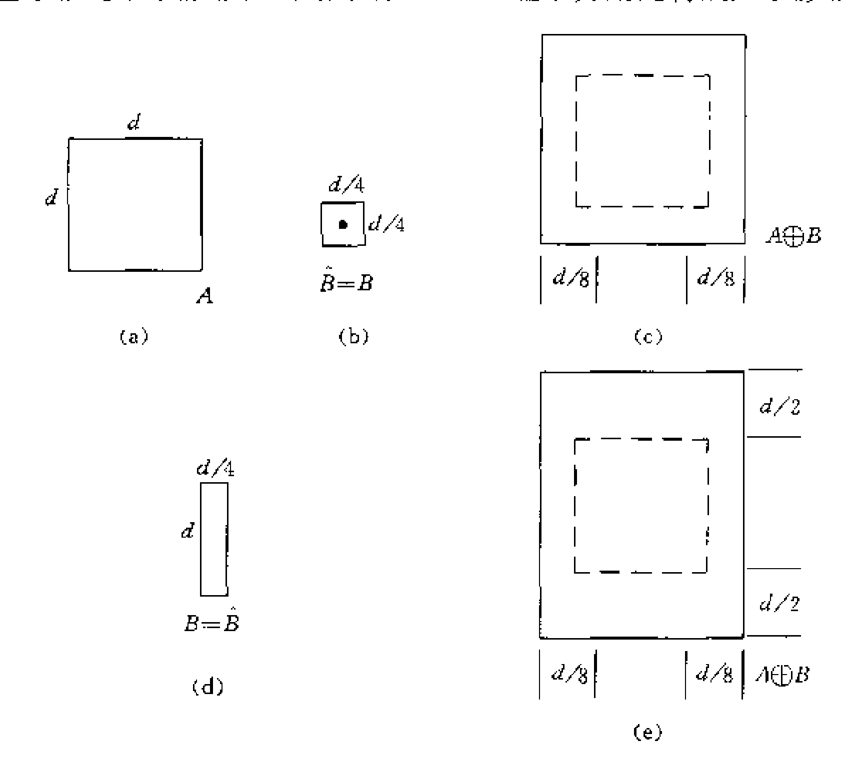


图 9-3 膨胀操作的例子

3. 腐蚀

A,B 为 Z^2 中的集合，A 被 B 腐蚀，记为 $A\ominus B$，其定义为

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \tag{9-14}$$

也就是说，A 被 B 的腐蚀的结果为所有使 B 被 x 平移后包含于 A 的点 x 的集合。与膨胀一样，式(9-11)也可以用相关的概念加以理解。

图 9-4 表示了类似于图 9-3 的一个过程。像以前一样，集合 A 在图 9-4(c)用虚线表示作为参考。实线表示若 B 的原点平移至 x 点超过此界限，则 A 不能完全包含 B。这样，在这个实线边界内的点构成了 A 被 B 的腐蚀。图 9-4(d)给出了伸长的结构元素，图 9-4(e)显示了 A 被此元素腐蚀的结果。注意原来的集合被腐蚀成一条线了。

膨胀和腐蚀是关于集合补和反转的对偶。也就是，

$$(A \ominus B)^c = A^c \oplus \hat{B} \tag{9-15}$$

关于上式的正确性可证明如下：

从腐蚀的定义可知，

$$(A \ominus B)^c = \{x \mid (B)_x \subseteq A\}^c$$

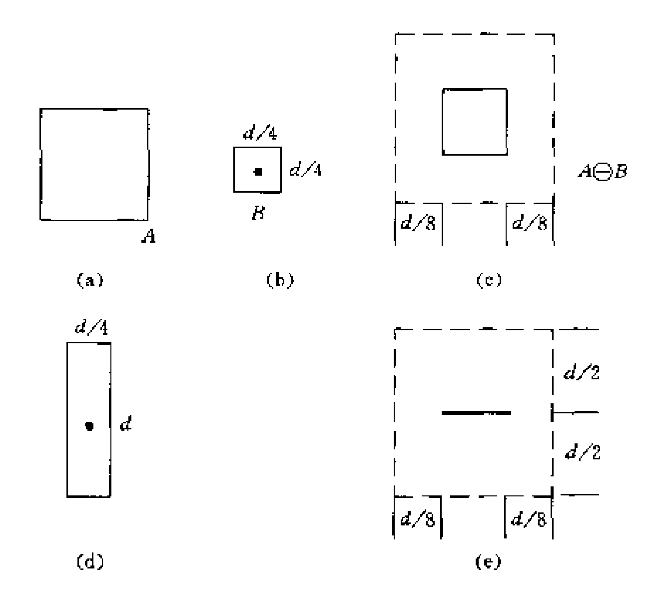


图 9-4　腐蚀操作的例子

如果集合$(B)_x$包含于集合 A，那么$(B)_x \cap A^c = \varnothing$，在这种情况下，上式变为

$$(A \ominus B)^c = \{x \mid (B)_x \cap A^c = \varnothing\}^c$$

但是满足$(B)_x \cap A^c = \varnothing$的集合 x 的补集是使$(B)_x \cap A^c \neq \varnothing$的 x 集合。这样，

$$\begin{aligned}(A \ominus B)^c &= \{x \mid (B)_x \cap A^c \neq \varnothing\} \\ &= A^c \oplus \hat{B}\end{aligned}$$

命题得证。

膨胀和腐蚀运算的一些性质对设计形态学算法进行图像处理和分析是非常有用的，下面列出几个较重要的性质：

① 交换性
$$A \oplus B = B \oplus A \tag{9-16}$$

② 结合性
$$A \oplus (B \oplus C) = (A \oplus B) \oplus C \tag{9-17}$$

③ 递增性
$$A \subseteq B \Rightarrow A \oplus C \subseteq B \oplus C \tag{9-18}$$
$$A \subseteq B \Rightarrow A \ominus C \subseteq B \ominus C$$

④ 分配性
$$(A \cup B) \oplus C = (A \oplus C) \cup (B \oplus C) \tag{9-19}$$
$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C) \tag{9-20}$$
$$A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C) \tag{9-21}$$
$$(B \cap C) \ominus A = (B \ominus A) \cap (C \ominus A) \tag{9-22}$$

这些性质的重要性是显而易见的。如分配性，如果用一个复杂的结构元素对图像作膨胀运算，则可以把这个复杂结构元分解为几个简单的结构元素的并集，然后，用几个简单的结构元素对图像分别进行膨胀运算，最后将结果再作并集运算，这样一来就可以大大简化运算的复杂性。

4．开运算(Opening)和闭运算(Closing)

如前边所见，膨胀扩大图像，腐蚀收缩图像。另外两个重要的形态运算是开运算和闭运算。开运算一般能平滑图像的轮廓，削弱狭窄的部分，去掉细的突出。闭运算也是平滑图像的轮廓，与开运算相反，它一般熔合窄的缺口和细长的弯口，去掉小洞，填补轮廓上的缝隙。

设 A 是原始图像，B 是结构元素图像，则集合 A 被结构元素 B 作开运算，记为 $A \circ B$，其

定义为

$$A \circ B = (A \ominus B) \oplus B \tag{9-23}$$

换句话说，A 被 B 开运算就是 A 被 B 腐蚀后的结果再被 B 膨胀。

设 A 是原始图像，B 是结构元素图像，集合 A 被结构元素 B 作闭运算，记为 $A \cdot B$，其定义为

$$A \cdot B = (A \oplus B) \ominus B \tag{9-24}$$

换句话说，A 被 B 作闭运算就是 A 被 B 膨胀后的结果再被 B 腐蚀。

图 9-5 图释了集合 A 被一个圆盘形结构元素作开运算和闭运算的情况。图 9-5(a)是集合 A，9-5(b)示出了在腐蚀过程中圆盘结构元素的各个位置，当完成这一过程时，形成分开的两个图形示于图 9-5(c)。注意，A 的两个主要部分之间的桥梁被去掉了。"桥"的宽度小于结构元素的直径；也就是结构元素不能完全包含于集合 A 的这一部分，这样就违反了式(9-14)的条件。由于同样的原因，A 的最右边的部分也被切除掉了。图 9-5(d)给出了对腐蚀的结果进行膨胀的过程，而图 9-5(e)示出了开运算的最后结果。同样地，图 9-5(f)～9-5(i)示出了用同样的结构元素对 A 作闭运算的结果。结果是去掉了 A 的左边对于 B 来说是较小的弯。注意，用一个圆形的结构元素对集合 A 作开运算和闭运算均使 A 的一些部分平滑了。

开运算和闭运算有一个简单的几何解释。假设我们把圆盘形结构元素 B 看作一个(平面的)"滚动球"。$A \circ B$ 的边界为 B 在 A 内滚动所能达到的最远处的 B 的边界所构成。这个解释能从图 9-5(a)得到图 9-5(e)。注意所有的朝外的突出角均被圆滑了，而朝内的则没有影响。突出的不能容下这球的部分被去掉。这种开运算的几何拟合性得出了集合论的一个定理：

A 被 B 的开运算就是 B 在 A 内的平移(保证$(B)_z \subseteq A$)所得到的集合的并集。这样开运算可以被描述为拟合过程，即：

$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\} \tag{9-25}$$

图 9-6 图释了这个概念，为了多样性这里我们用了一个非圆形的结构元素。

闭运算也有类似的几何解释。再次用滚动球作例子，只不过我们在边界外边滚动该球(开运算和闭运算是对偶的，所以让小球在外面滚动是合理的)。有了这种解释，图 9-5(i)就很容易从图 9-5(a)得到。注意所有的朝内的突出角均被圆滑了，而朝外的则保持不变。集合 A 的最左边的凹入被大幅度减弱了。几何上，点 Z 为 $A \cdot B$ 的一个元素，当且仅当包含 Z 的$(B)_z$ 与 A 的交集非空，即$(B)_z \cap A \neq \varnothing$。图 9-7 解释了这一性质。

像膨胀和腐蚀一样，开运算和闭运算是关于集合补和反转的对偶。也就是

$$(A \cdot B)^c = (A^c \circ \hat{B}) \tag{9-26}$$

开运算有下列性质：

① $A \circ B$ 是集合 A 的子集(子图)；

② 如果 C 是 D 的子集，则 $C \circ B$ 是 $D \circ B$ 的子集；

③ $(A \circ B) \circ B = A \circ B$。

同样，闭运算有下列性质：

① A 是集合 $A \cdot B$ 的子集(子图)；

② 如果 C 是 D 的子集，则 $C \cdot B$ 是 $D \cdot B$ 的子集；

③ $(A \cdot B) \cdot B = A \cdot B$。

这些性质有助于对用开运算和闭运算构成的形态滤波器时所得到的结果的理解。例如，用开运算构造一个滤波器。我们参考上面的性质：

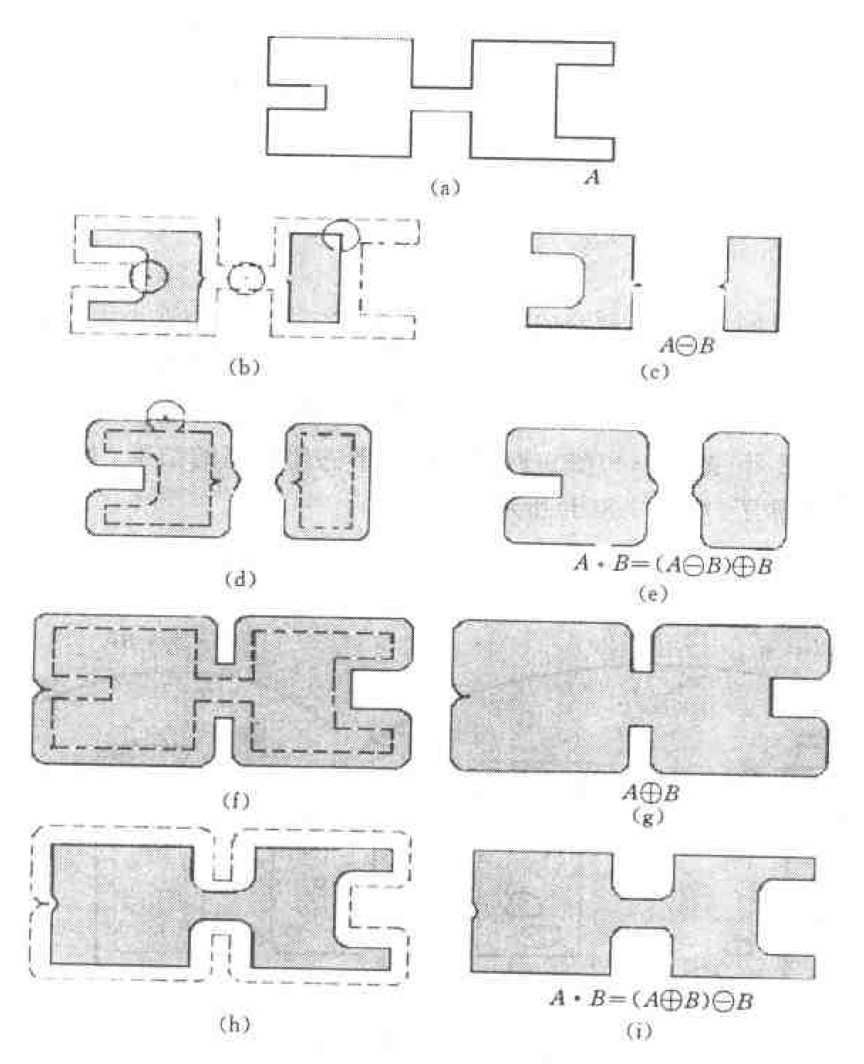


图 9-5 开运算和闭运算的图示

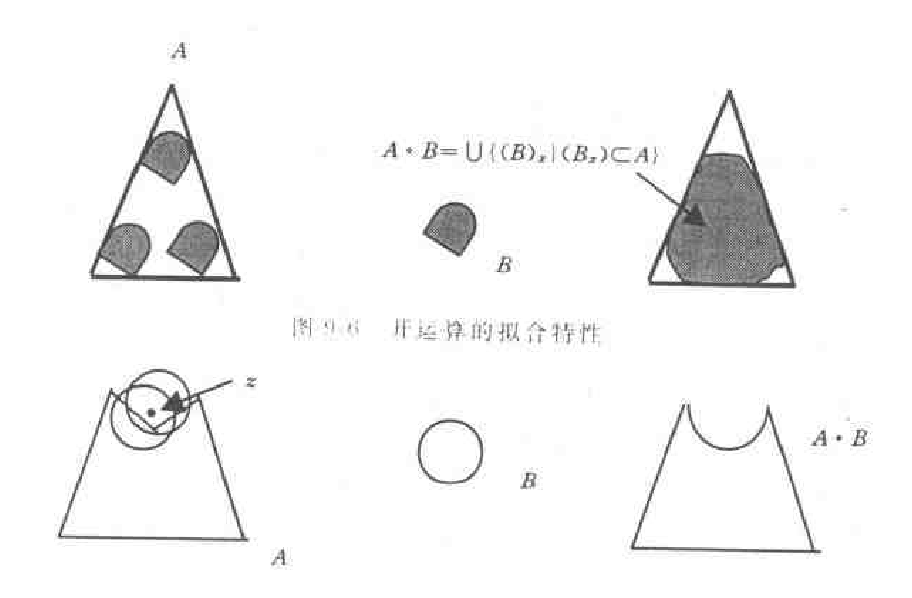


图 9-6 开运算的拟合特性

图 9-7 闭运算的几何解释

(i) 结果是输入的子集;(ii)单调性会被保持;(iii)多次同样的开运算对结果没有影响。最后一条性质有时称为幂等性。同样的解释适合于闭运算。

考虑图 9-8(a)的简单的二值图像,它包含一个被噪声影响的矩形目标。这里噪声用暗元素(阴影)在亮的背景表示,而光使暗目标为空的。注意集合 A 包含目标和背景噪声,而目标中的噪声构成了背景显示的内部边界。目的是去除噪声及其对目标的影响,并对目标的影响越小越好。形态"滤波器"$(A \circ B) \cdot B$ 可以用来达到此目的。图 9-8(c)显示了用一个比所有噪声成份都大的圆盘形结构元素对 A 进行开运算的结果。注意这步运算考虑了背景噪声但对内部边界没有影响。

因为在这个理想的例子中,所有的背景噪声成份的物理大小均小于结构元素,背景噪声在开运算的腐蚀过程中被消除(腐蚀要求结构元素完全包含于被腐蚀的集合内)。而目标内的噪声成份的大小却变大了(图 9-8(b)),这在意料之中,原因是目标中的空白事实上是内部边界,在腐蚀中会变大。最后,图 9-8(e)图 9-8(c)示出了形态闭运算的结果。内部的边界在闭运算后的膨胀运算中被消除了,如图 9-8(d)所示。

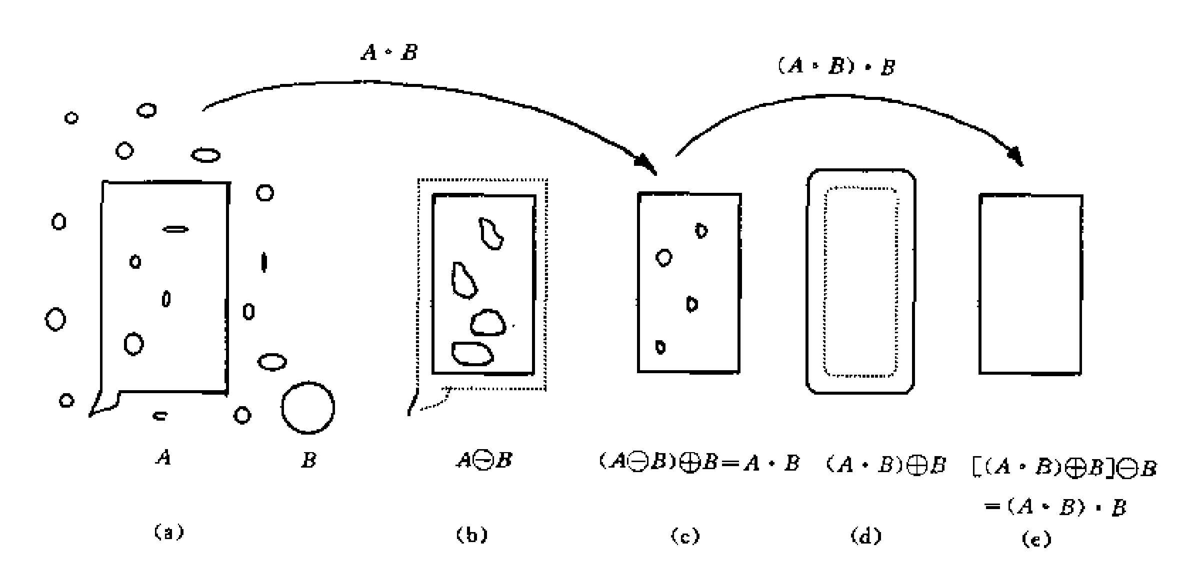


图 9-8　形态学滤波

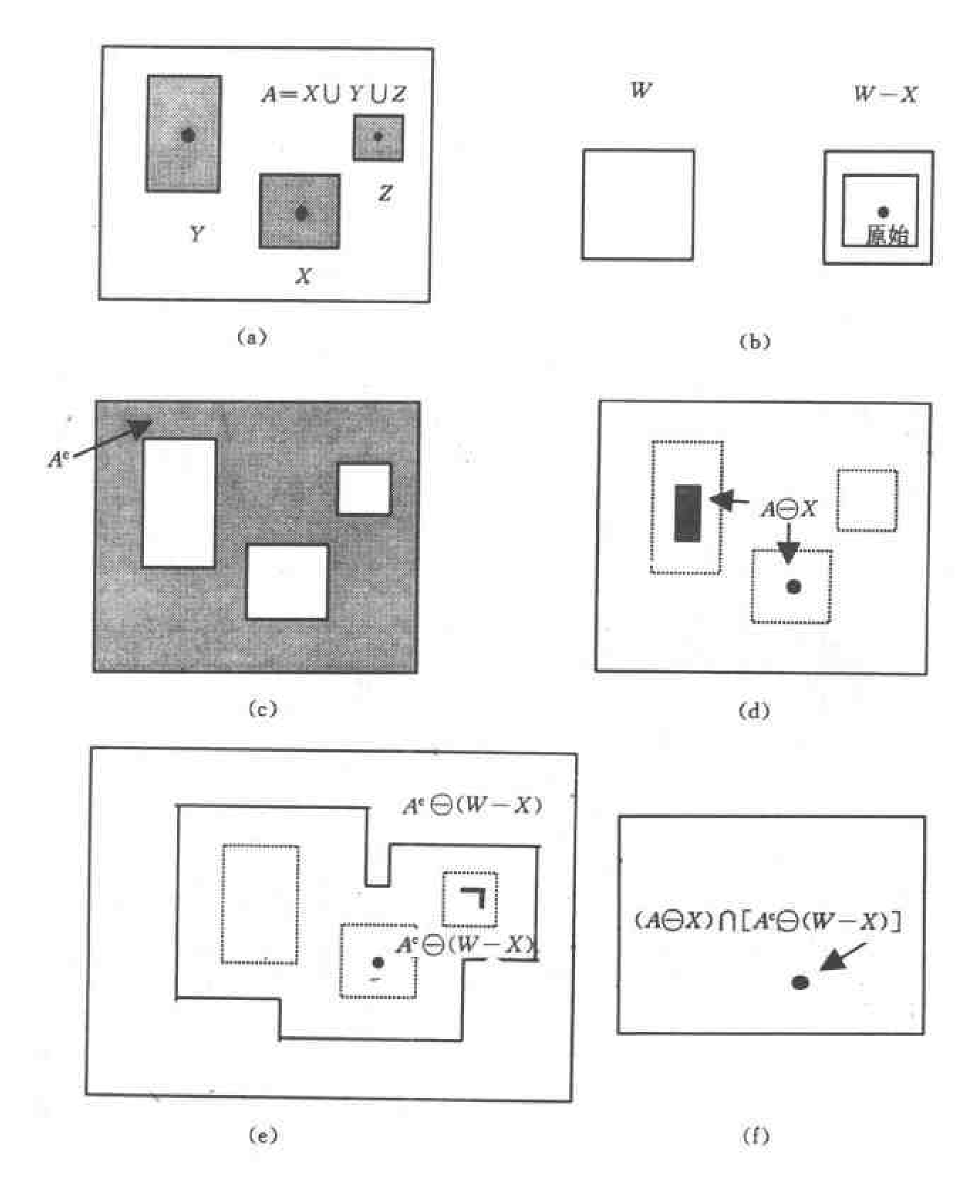
5. 击中(Hit)击不中(Miss)变换(HMT)

形态学中击中(Hit)击不中(Miss)变换是形状检测的基本工具。我们通过图 9-9 引入这个概念。图中集合 A 包含三个部分(子集),记为 X,Y,Z。图 9-9(a)~(c)中的图形为原始集合,而图 9-9(d)和(e)中的阴影为形态运算的结果。目标是找到一个图形 X 的位置。

让每个图形的原点位于它的重心。如果用一个小窗口 W 包含 X,X 关于 W 的本地背景是图 9-9(b)中的集合差$(W-X)$。图 9-9(c)为集合 A 的补。图 9-9(d)示出 A 被 X 腐蚀的结果。A 被 X 的腐蚀在 X 中只有 X 的原点,这样 X 才能完全包含于 A。图 9-9(e)表示集合 A 的补被本地背景集合$(W-X)$的腐蚀;外围阴影区域也是腐蚀结果的一部分。从图 9-9(d)和(e),可以看出集合 X 在集合 A 中的位置是 A 被 X 的腐蚀和 A^c 被$(W-X)$的腐蚀的交集,如图 9-9(f)所示。这个交集正是我们所要找的。换句话说,如果 B 记为由 X 和其背景构成的集合,B 在 A 中的匹配,记为 $A \circledast B$,则

$$A \circledast B = (A \ominus X) \cap [A^c \ominus (W - X)] \tag{9-27}$$

可以这样来概括这种表示法,让 $B=(B_1,B_2)$,其中 B_1 是由和目标相关的 B 的元素形成的集合,而 B_2 是由和相应的背景相关的 B 的元素集合。根据前面的讨论,$B_1=X$,$B_2=(W-$



(a) (b) (c) (d) (e) (f)

图 9-9 击中(Hit)击不中(Miss)变换图例

X)。用这种表示法,公式(9-27)变为

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2) \tag{9-28}$$

用集合差的定义及膨胀和腐蚀的对偶关系,也可以把公式(9-28)写为

$$A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2) \tag{9-29}$$

这样集合 $A \circledast B$ 包括所有的点,同时,B_1 在 A 中找到了一个匹配"击中",B_2 在 A^c 中找到了匹配"击中"。

9.3 一些基本形态学算法

在前面讨论的背景知识基础之上,我们可以探讨形态学的一些实际应用。当处理二值图像时,形态学的主要应用是提取表示和描述图像形状的有用成份。特别是用形态学方法提取某一区域的边界线、连接成份、骨骼、凸壳的算法是十分有效的。此外,区域填充、细化、加粗、裁剪等

处理方法也经常与上述算法相结合在预处理和后处理中使用。

为概念清楚起见，这些算法的讨论大部分采用的是二值图像，即只有黑和白两级灰度，1表示黑，0表示白。

9.3.1 边缘提取算法

集合 A 的边界记为 $\beta(A)$，可以通过下述算法提取边缘：设 B 是一个合适的结构元素，首先令 A 被 B 腐蚀，然后求集合 A 和它的腐蚀的差。如下式所示：

$$\beta(A) = A - (A \ominus B) \tag{9-30}$$

图 9-10 解释了边缘提取的过程。它表示了一个简单的二值图像，一个结构元素和用公式 9-30 得出的结果。图 9-10(b)中的结构元素是最常用的一种，但它决不是惟一的。如果采用一个 5×5 全"1"的结构元素，可得到一个二到三个像素宽的边缘。应注意的是，当集合 B 的原点处在集合的边界时，结构元素的一部分位于集合之外。这种条件下的通常的处理是约定集合边界外的值为 0。

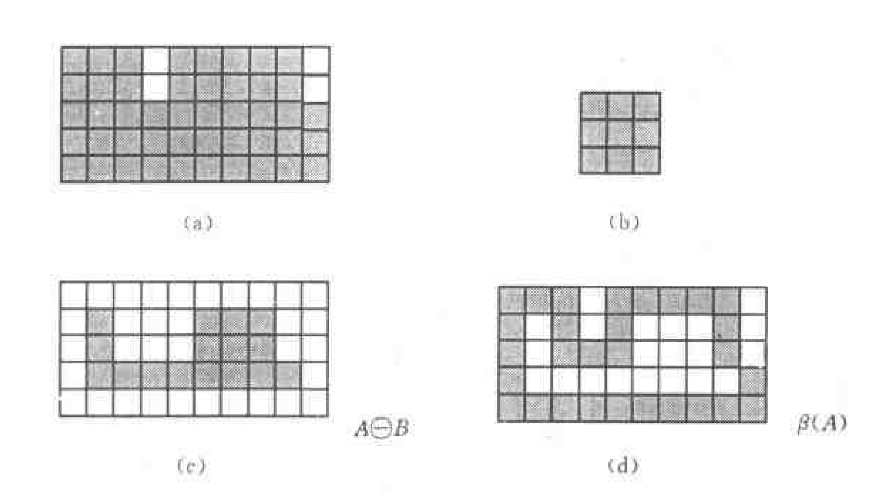


图 9-10 边缘提取算法示意图

9.3.2 区域填充算法

下面讨论的是一种基于集合膨胀，取补和取交的区域填充的简单的算法。在图 9-11 中，A 表示一个包含一个子集的集合，子集的元素为 8 字形的连接边界的区域。从边界内的一点 P 开始，目标是用 1 去填充整个区域。

假定所有的非边界元素均标为 0，我们把一个值 1 赋给 P 开始这个过程。下述过程将把这个区域用 1 来填充：

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,\cdots \tag{9-31}$$

其中，$X_0 = P$，B 为对称结构元素，如图 9-11(c)所示。当 k 迭代到 $X_k = X_{k-1}$ 时，算法终止。集合 X_k 和 A 的并集包括填充的集合和边界。

如果公式(9-31)的膨胀过程一直进行，它将填满整个区域。然而，每一步与 A^c 的交把结果限制在我们感兴趣的区域内(这种限制过程有时称为条件膨胀)。图 9-11 剩下的部分解释了公式(9-31)的进一步技巧。尽管这个例子只有一个子集，只要每个边界内给一个点，这个概念可清楚地用在任何有限个这样的子集中。

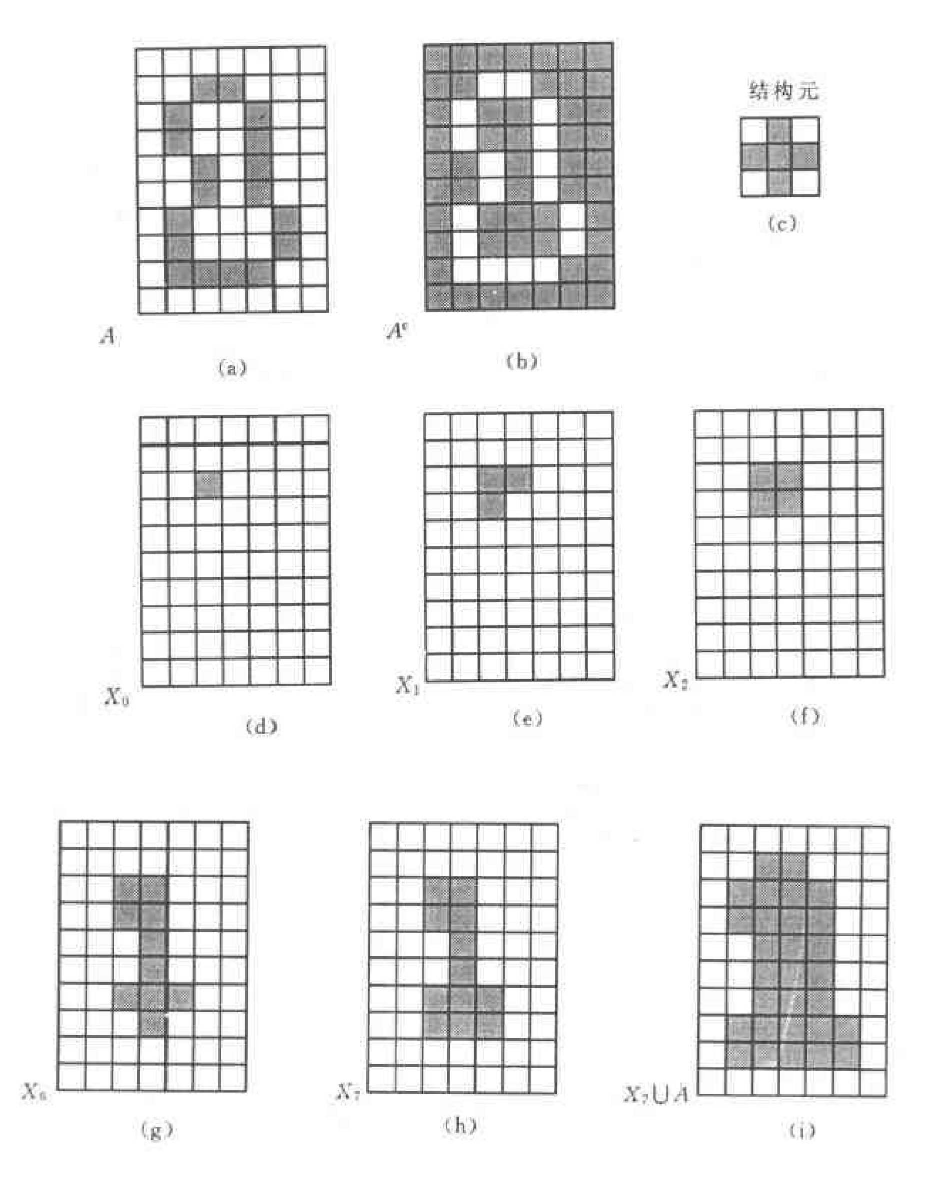


图 9-11 区域填充算法

9.3.3 连接部分提取算法

在实际应用中，在二值图像中提取相连接部分是许多自动图像分析应用所关注的问题。Y 表示一个包含于集合 A 相连接部分，假设 Y 内的一个点 P 已知。那么下述迭代表达式可得到 Y 中的所有元素：

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1,2,\cdots \tag{9-32}$$

其中 $X_0=P$，B 为一合适的结构元素，如图 9-12 所示。如果 $X_k=X_{k-1}$ 则算法收敛，并使 $Y=X_k$。

公式(9-32)在形式上与(9-31)相似。惟一的不同是用 A 代替了 A^c，这是因为所提取的全部元素(也就是，相连组成部分的元素)均标记为 1。每一迭代步和 A 求交集可除去以标记为 0 的元素为中心的膨胀。图 9-12 图释了公式(9-32)的操作技巧。这里，结构元素的形状是 8 连接的，与区域填充算法一样，以上讨论的结果可以应用于任何有限的包含在集合 A 中的连接部分。图中(a)集 A 包含一个连接部分 Y 和初始点 P；(b)是结构元；(c)是第一次迭代结果；

(d)是第二次迭代结果；(e)是最终结果。

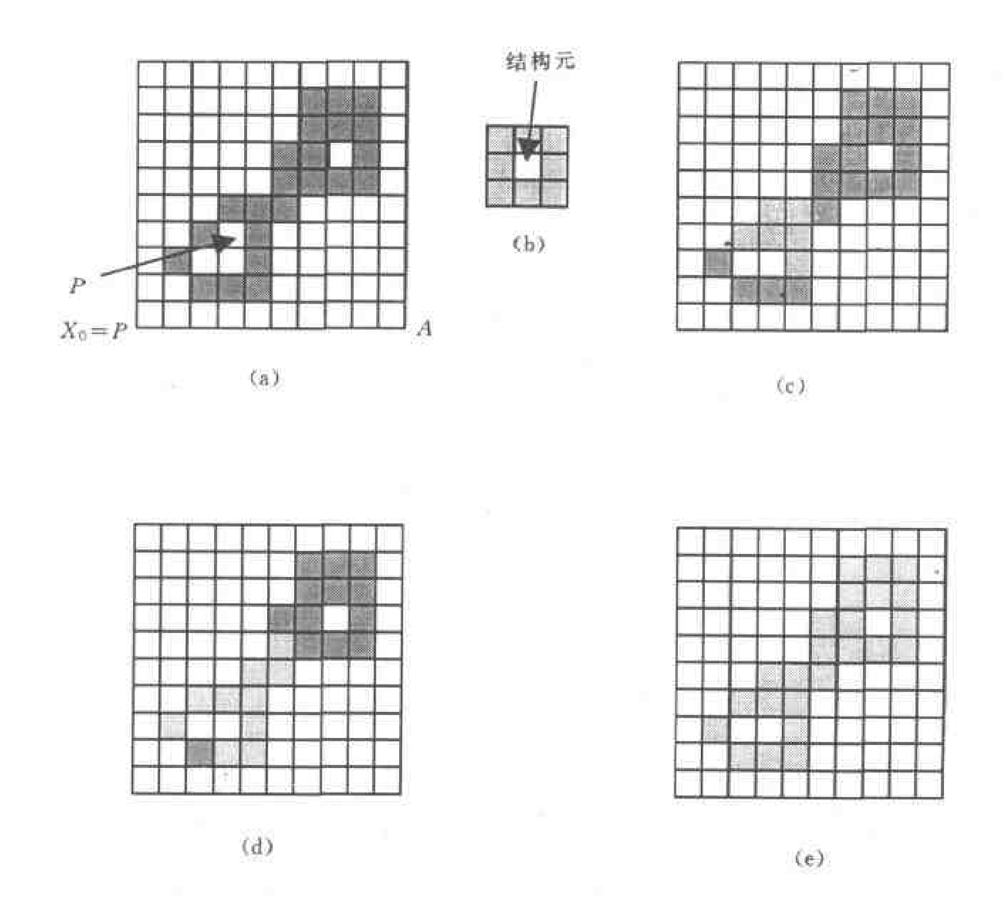


图 9-12　连接部分提取算法

9.3.4　凸壳算法

集合的凸壳是一个有用的图像描述工具。在此，我们提出一种获得集合 A 凸壳 $C(A)$ 的简单形态学算法。设 B^i，$i=1,2,3,4$，代表 4 个结构元素。这个处理过程由下述公式实现：

$$X_k^i=(X_{k-1}^i \circledast B^i)\cup A \qquad i=1,2,3,4 \qquad k=1,2,\cdots \tag{9-33}$$

其中 $X_0^i=A$。现令 $D^i=X_{\text{conv}}^i$，下标 conv 表示当 $X_k^i=X_{k-1}^i$ 时收敛。那么，A 的凸壳为

$$C(A)=\bigcup_{i=1}^{4}D^i \tag{9-34}$$

换句话说，这个过程包括对 A 和 B^1 重复使用击中(hit)或击不中(miss)变换；当没有进一步的变化发生时，求 A 和所谓的结果 D^1 并集。对 B^2 重复此过程直到没有进一步的变化为止。4 个结果 D 的并构成了 A 的凸壳。

图 9-13 解释了公式(9-33)和(9-34)的过程。图 9-13(a)示出了为提取凸壳的结构元素(每个结构元素的原点位于它的中心)。图 9-13(b)给出了要提取凸壳的集合 A，从 $X_0^1=A$ 开始，重复公式(9-33)四步后得到的结果如图 9-13(c)所示。然后令 $X_0^2=A$ 再次利用公式(9-33)得到的结果示于图 9-13(d)(注意只用两步就收敛了)。下两个结果用同样的方法得到。最后，把图 9-13(c)、(d)、(e)和(f)中的集合求并的结果就为所求凸壳。每个结构元素对结果的贡献在图 9-13(h)的合成集合中用不同加亮表示。

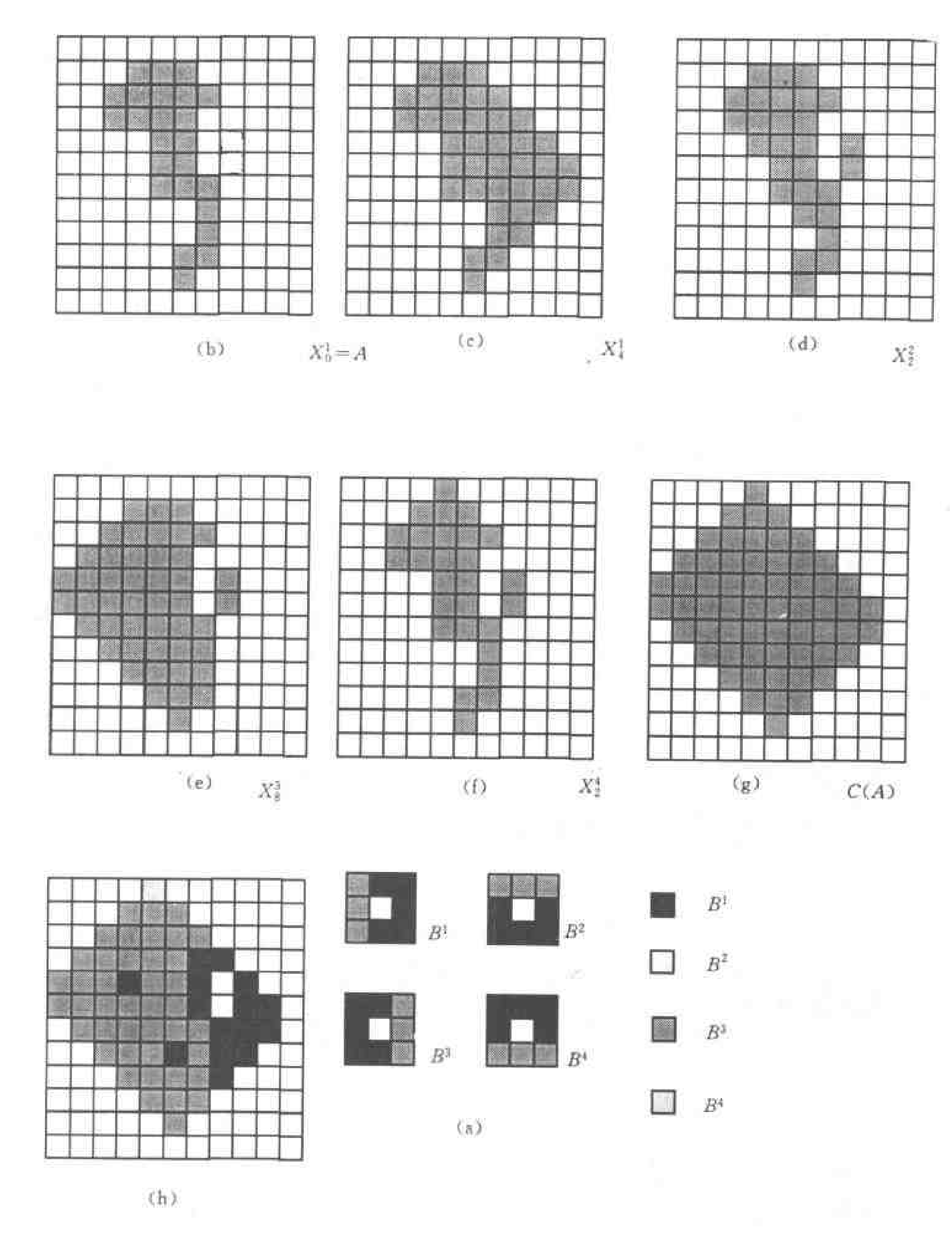


图 9-13　凸壳算法示例

9.3.5　细化

集合 A 被结构元素的细化用 $A \otimes B$ 表示，根据击中(hit)(或击不中 miss)变换定义：

$$\begin{aligned} A \otimes B &= A - (A \circledast B) \\ &= A \cap (A \circledast B)^c \end{aligned} \tag{9-35}$$

对称细化 A 的一个更有用的表达是基于结构元素序列：

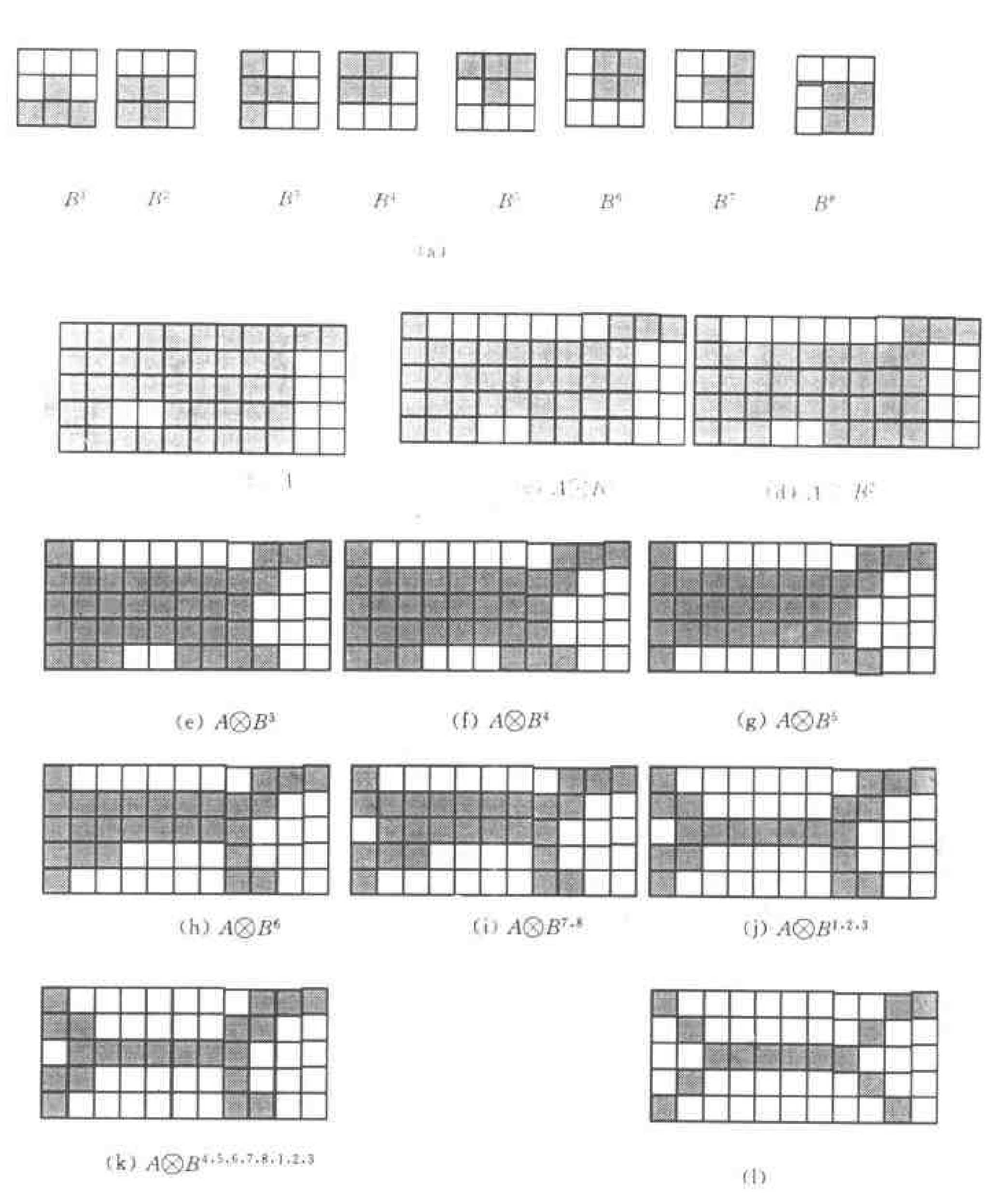
$$\{B\} = \{B^1, B^2, \cdots, B^n\} \tag{9-36}$$

其中 B^i 是 B^{i-1} 的旋转。根据这个概念，我们现定义被一个结构元素序列的细化为

$$A \otimes \{B\} = ((\cdots((A \otimes B^1) \otimes B^2)\cdots) \otimes B^n) \tag{9-37}$$

换句话说，这个过程是用 B^1 细化 A，然后用 B^2 细化前一步细化的结果等等，直到 A 被 B^n 细化。整个过程重复进行到没有进一步的变化发生为止。

图 9-14(a)是一组用于细化的结构元素，图 9-14(b)为用上述方法细化的集合 A。图 9-14(c)示出用 B^1 细化 A 得到的结果，图 9-14(d)～(k)为用其他结构元素细化的结果。当第二次通过 B^4 时收敛。图 9-14(k)示出细化的结果。



(e) $A\otimes B^3$　(f) $A\otimes B^4$　(g) $A\otimes B^5$

(h) $A\otimes B^6$　(i) $A\otimes B^{7,8}$　(j) $A\otimes B^{1,2,3}$

(k) $A\otimes B^{4,5,6,7,8,1,2,3}$　(l)

图 9-14　细化处理

9.3.6　粗化算法

粗化是细化的形态学上的对偶，记为 $A\odot B$，定义为

$$A\odot B = A \cup (A \circledast B) \tag{9-38}$$

其中 B 是适合粗化的结构元素。像细化一样，粗化可以定义为一个序列运算：

$$A\odot\{B\} = (\cdots((A \odot B^1)\odot B^2)\cdots)\odot B^n \tag{9-39}$$

用来粗化的结构元素同细化的结构元素具有相同的形式。只是所有的 0 和 1 交换位置。然而，在实际中，粗化的算法很少使用。相反的，通常的过程是细化集合的背景，然后求细化结果的补而达到粗化的结果。换句话说，为了粗化集合 A，我们先令 $C=A^c$，细化 C，然后得到 C^c 即为粗化结果。图 9-15 解释了这个过程。

如图 9-15(d)所示，这个过程可能产生一些不连贯的点，这取决于 A 的性质。因此，用这种方法粗化通常要进行一个简单的后处理步骤来清除不连贯的点。从图 9-15(c)可以看出，细化的背景为粗化过程形成一个边界。这个有用的性质在直接使用公式(9-39)实现粗化过程中不会出现，这是用背景细化来实现粗化的一个主要原因。

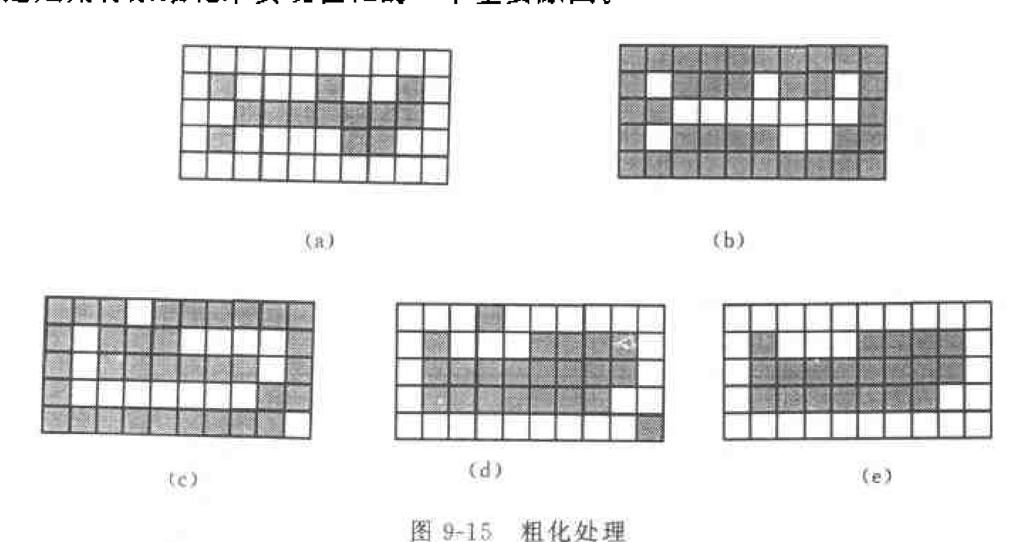


图 9-15 粗化处理

9.3.7 骨骼化算法

利用形态学方法提取一个区域的骨骼可以用腐蚀和开运算表示。也就是，A 的骨骼记为 $S(A)$，骨骼化可以表示如下：

$$S(A)=\bigcup_{k=0}^{K}S_k(A) \tag{9-40}$$

和

$$S(A)=\bigcup_{k=0}^{K}\{(A\ominus kB)-[(A\ominus kB)\circ B]\} \tag{9-41}$$

其中 B 是结构元素，$A\ominus kB$ 表示对 A 连续腐蚀 k 次，就是

$$A\ominus kB=((\cdots(A\ominus B)\ominus B)\ominus\cdots)\ominus B$$

共执行 k 次，K 是 A 被腐蚀为空集以前的最后一次迭代的步骤，即

$$K=\max\{k\mid(A\ominus kB)\neq\varnothing\} \tag{9-42}$$

等式(9-40)和等式(9-41)明确表明集合 A 的骨骼 $S(A)$ 可以由骨骼子集 $S_k(A)$ 的并得到，以上等式同样表明 A 可以通过等式(9-42)从这些子集中重构：

$$A=\bigcup_{k=0}^{K}(S_k(A)\oplus kB) \tag{9-43}$$

公式中 $(S_k(A)\oplus kB)$ 表明参数 k 是对子集 $S_k(A)$ 连续膨胀 k 次。正如前面所述，它相当于下式：

$$(S_k(A)\oplus kB)=((\cdots(S_k(A)\oplus B)\oplus B\cdots)\oplus B \tag{9-44}$$

图 9-16 的解释说明了以上讨论的概念。第一列显示了原始集合(顶部)和通过结构元素 B 两次腐蚀的图形。由于再多一次对 A 的腐蚀将产生空集，所以选取 $K=2$。第二列显示了第一列通过 B 的开运算而得到的图形。以上结果可以通过以前讨论过的开运算拟合性质加以解释。第

三列仅仅显示出第一列与第二列的差别。第四列包含两个部分骨骼及最后的结果(第四列的底部)。最后的骨骼不但比所要求的更粗,而且相比较更重要,它是不连续的。形态学给出了就特定图形侵蚀和空缺的描述。通常,骨骼必须最大限度的细化、相连、最小限度的腐蚀。第五列显示了$S_0(A)$、$S_1(A)\oplus B$以及$(S_2(A)\oplus 2B)=(S_2(A)\oplus B)\oplus B$。最后一列显示了图像$A$的重构。由公式(9-42)可知,$A$就是第五列中膨胀骨骼子集的"并"。

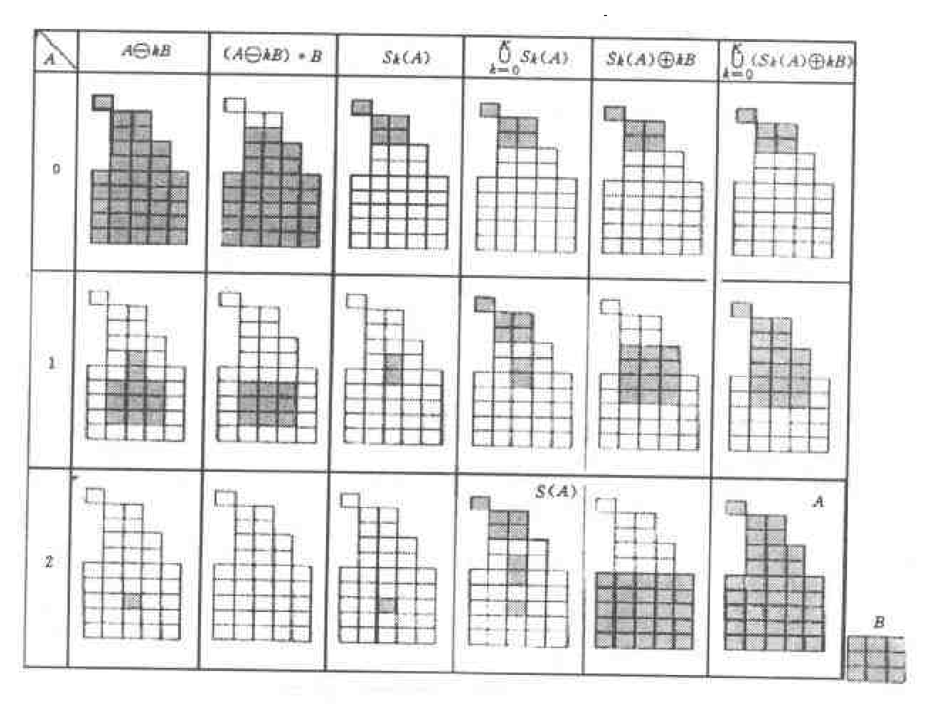


图 9-16 骨骼化处理结果

9.3.8 裁剪

由于图形细化和骨骼化运算法有可能残留需要在后续处理中去除的寄生成份,因而剪贴方法成为对图形细化、骨骼化运算的必要补充。下面将讨论裁剪问题,我们将运用已成熟的理论来阐明如何通过融合现今已有的技术来解决这样的一个问题。

分析每个待识别字符的骨骼形状是自动识别手写字符的一种常见处理方法。由于对组成字符的笔画的不均匀腐蚀,字符的骨架常常带有"毛刺"(一种寄生成份)。这里将提出一种解决这种问题的形态学方法。首先我们假设寄生成份"毛刺"的长度不超过3个像素。

图9-17(a)显示了手写字符"a"的骨骼。在字符最左边部分的寄生成份是一种我们感兴趣的典型的待去除成份。去除的方法是基于不断减少该字符的终点,对寄生成份加以抑制。当然不可否认这样也不可避免的会消去(或减少)被处理字符其余必要的骨架,但是缺少的结构信息是在我们最多不超过3个像素的假设前提下,即最多减少3个像素的字符结构信息的前提下。对于一个输入集合A,通过一系列用于检测字符端点的结构元素的细化处理,达到我们所希望的结果。即

$$X_1 = A \otimes \{B\} \tag{9-45}$$

等式(9-45)中$\{B\}$表示在图9-17(b)和(c)中的结构元序列。结构元素的序列包含两个不同的结构,每一个结构将对全部8个元素作90°的旋转,图9-17(b)中的"×"表示一个"不用考虑"的情况,在某种意义上,不管该位置上的值是0还是1都毫无关系。许多图形学文献记载的结

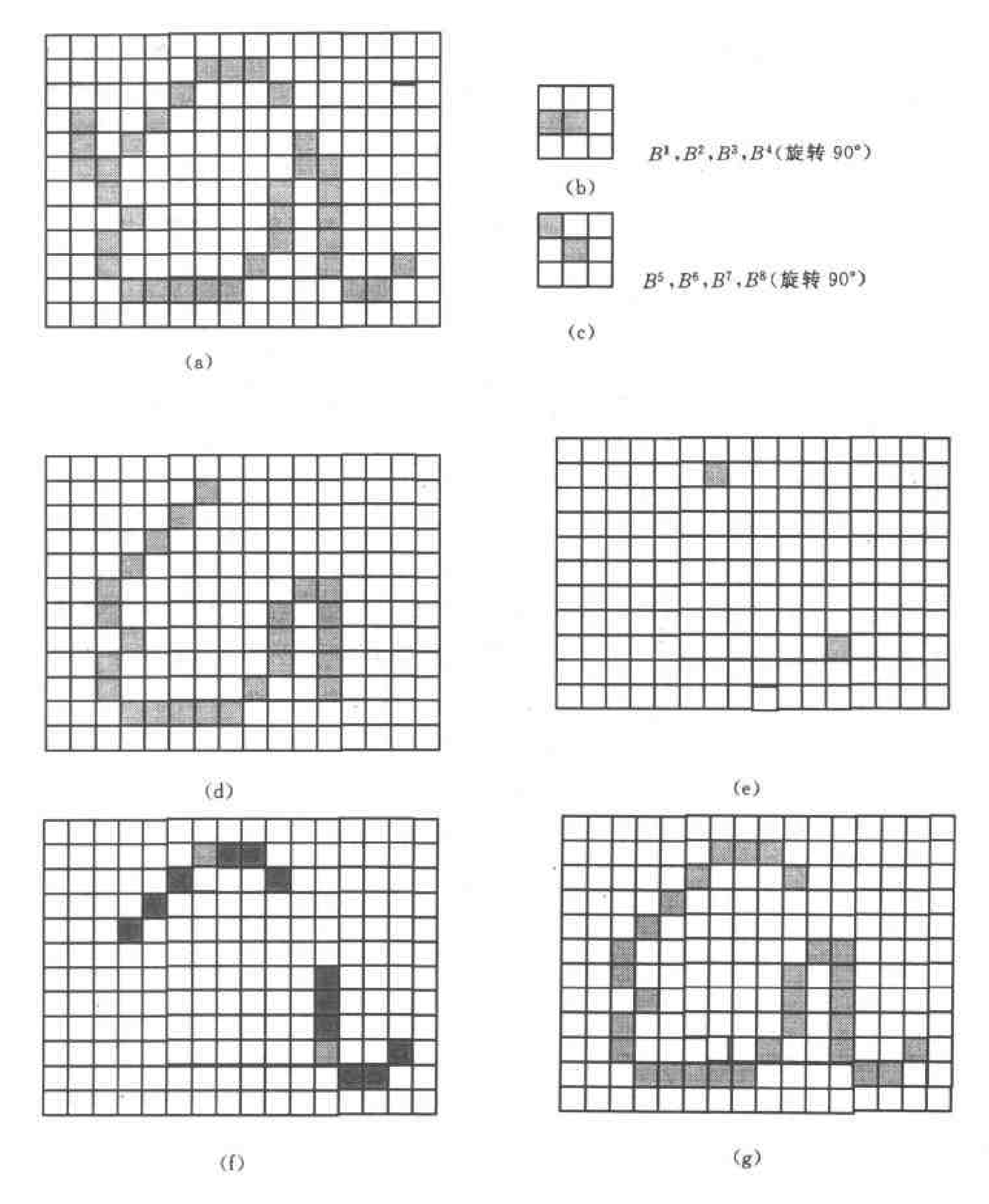


图 9-17 裁剪的例子

(a)是原像，(b)和(c)是结构元素，(d)细化三次的结果，(e)端点，

(f)在(a)的条件下端点的膨胀，(g)裁剪后的图像

果都是基于类似于图 9-17(b)中单一结构的运用基础之上的，不过不同的是，在第一列中多了"不用考虑"的状态而已。这样的处理是不完善的。例如，这个元素将标识图 9-17(a)位于第 8 排，第四列作为最后一点的点，如果减去该元素将破坏这一笔的连接性。

连续对 A 运用等式(9-45)三次将生成图 9-17(d)中的集合 X_1。下一步将是把字符"恢复"到最初的形状，同时将寄生的成份去除。这首先需要建立包含图 9-17(e)所有边缘信息的集合 X_2，

$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k) \tag{9-46}$$

等式(9-46)中 B^t 是和前面一样的端点检测因子，下一步对边缘进行三次放大处理，集合 A 作为消减因子：

$$X_3 = (X_2 \oplus H) \cap A \tag{9-47}$$

等式(9-47)中 H 是一个值为 1 的 3×3 的结构元素，类似局域填充和连接成份的提取的情况，这一类条件膨胀处理有效的避免了在我们感兴趣区域外值 1 元素的产生。正如图 9-17(f)中显示的结果一样。最后，X_3 和 X_1 的并生成了最后的结果：

$$X_4 = X_1 \cup X_3 \tag{9-48}$$

正如图 9-17(g)中所示。

在更复杂的情况下，使用公式(9-47)有时可以捡拾一些寄生分枝的“尖端”。如果分支端点离骨骼较近时，这种情况便会发生。尽管可以通过等式(9-45)减少，但是由于它们是 A 中的有效点而在膨胀处理中再次出现。除非只有所有的寄生元素再次获得的情况下(当这些寄生元素与字符笔画相比不够长时，这将是一种出现机率非常少的情况)，如果寄生元素处在非连接区域，那末检测和减少寄生元素才会变得容易一些。

在这一点上一种自然而然的想法就是必须有一种方法来解决这个问题。例如，我们可以通过运用公式(9-45)，仅仅对被删除点进行跟踪和对所有的留下的端点进行再连接。这样的选择是正确的，它的优点是使用简单的形态结构来解决所有的问题。在实际情况中这一套工具是可用的情况下，它的优点在于不必再写新的算法。我们所作的仅仅是简单地把形态函数组合到算子序列中就可以了。

表 9-1 总结了前边讨论的数学形态学算法及其结果，图 9-18 示出了所使用的基本结构元素。

表 9-1　形态学结论和特性的总结

操作	等　　式	评　　述
平移	$(A)_x=\{c\mid c=a+x, a\in A\}$	把 A 的原点平移到 x 点
映射	$\hat{A}=\{x\mid x=-b, b\in A\}$	相对于原点映射关于集合 A 的所有元素
补集	$A^c=\{x\mid x\notin A\}$	所有不属于 A 的点集
差集	$A-B=\{x\mid x\in A, x\notin B\}=A\cap B^c$ 属于 A 但不属于 B 的点集	
膨胀	$A\oplus B=\{x\mid[(\hat{B})_x\cap A]\neq\varnothing\}$	“扩展”A 的边界
腐蚀	$A\ominus B=\{x\mid(B)_x\subseteq A\}$	“收缩”A 的边界
开	$A\circ B=(A\ominus B)\oplus B$	平滑轮廓，切除狭区，去除小的孤岛及突刺
闭	$A\bullet B=(A\oplus B)\ominus B$	平滑轮廓，连接小狭区，纵向细化沟渠并且去除小洞
击中和击不中变换	$A\circledast B=(A\ominus B_1)\cap(A^c\ominus B_2)$ $=(A\ominus B_1)-(A\oplus\hat{B}_2)$	在一个点集上，B_1 在 A 中，B_2 在 A^c 中同时找到了匹配点
边缘提取	$\beta(A)=A-(A\ominus B)$	提取集合 A 的边界上的点集
区域填充	$X_k=(X_{k-1}\oplus B)\cap A^c$ $X_0=p$　$k=1,2,\cdots$	给定 A 的一个区域中的一点 p，填充这个区域

续表

操作	等　　式	评　述
连接的部分	$X_k=(X_{k-1}\oplus B)\cap A$； $X_0=p\quad k=1,2,\cdots$	给定 A 中一个连接的部分 Y 中的一点 p，提取 Y
凸壳	$X_k^i=(X_{k-1}^i\circledast B^i)\cup A$；$i=1,2,3,4$ $k=1,2,\cdots\quad X_0^i=A$ $D^i=X_{conv}^i\quad C(A)=\bigcup_{i=1}^{4}D^i$	找到集合 A 的凸壳 $C(A)$，其中 conv 在 $X_k^i=X_{k-1}^i$ 的意义上表示收敛
细化	$A\otimes B=A-(A\circledast B)$ $=A\cap(A\circledast B)^c$ $A\otimes\{B\}=((\cdots((A\otimes B^1)\otimes B^2)\cdots)\otimes B^n)$ $\{B\}=\{B^1,B^2,\cdots,B^n\}$	细化集合 A。前两个等式给出了细化的基本定义。后两个等式表示用一系列结构元素来进行的细化
粗化	$A\odot B=A\cup(A\circledast B)$ $A\odot\{B\}=(\cdots((A\odot B^1)\odot B^2)\cdots)\odot B^n$	粗化集合 A。(参阅以前关于结构元素序列的部分)。把细化中的 0 和 1 调换来使用即可粗化
骨骼	$S(A)=\bigcup_{k=0}^{K}S_k(A)$ $S_k(A)=\bigcup_{k=0}^{K}\{(A\ominus kB)-[(A\ominus kB)\circ B]\}$ $A=\bigcup_{k=0}^{K}(S_k(A)\oplus kB)$	找到集合 A 的骨骼 $S(A)$。最后一个公式表明 A 可由它的骨骼子集 $S_k(A)$ 重构。在所有等式中，K 是迭代步骤的次数，超过 K 次集合 A 被腐蚀为空集。符号 $(A\ominus kB)$ 表示连续腐蚀 k 次的迭代
裁剪	$X_1=A\otimes\{B\}$ $X_2=\bigcup_{k=1}^{K}(X_1\circledast B^k)$ $X_3=(X_2\oplus H)\cap A$ $X_4=X_1\cup X_3$	X_4 是裁剪集合 A 后的结果。用于获得 X_1 的第一等式的使用次数必须指定。等式分别使用了前边介绍过的结构元素

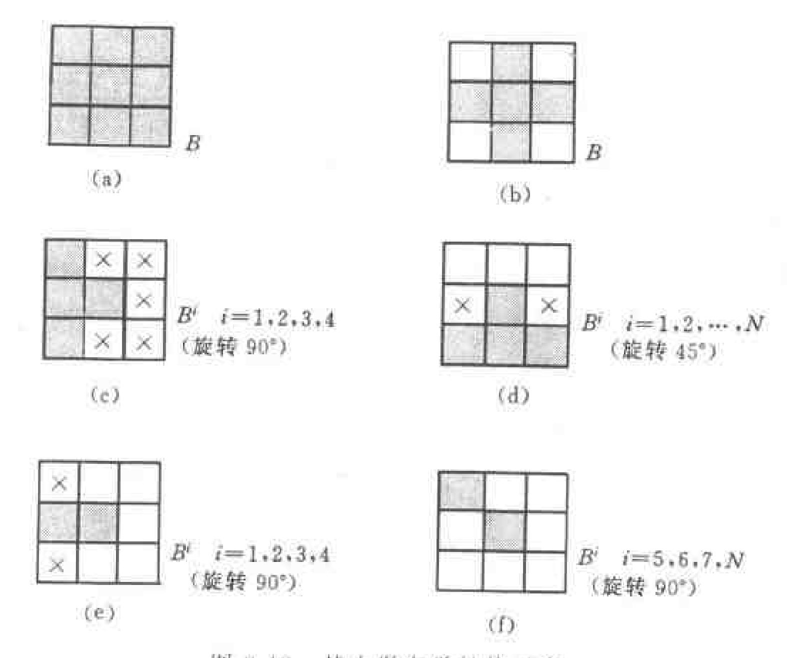


图 9-18　基本形态学结构元素

9.4 灰度图像的形态学处理

前边针对二值图像的形态学处理的基本运算作了系统的介绍，这些基本算法可方便地推广至灰度图像的处理。这一节我们将讨论对灰度图像的基本处理，即：膨胀、腐蚀、开运算、闭运算。由此建立一些基本的灰度形态运算法则。与前边相同，这一节的重点是运用灰度形态学提取描述和表示图像的有用成份。特别是，我们将通过形态学梯度算子开发一种边缘提取和基于纹理的区域分割算法。同时，我们将讨论在预处理及后处理步骤中非常有用的平滑及增强处理算法。

与前边二值图像形态学处理理论不同的是在以下的讨论中我们将处理数字图像函数而不是集合。设 $f(x,y)$ 是输入图像，$b(x,y)$ 是结构元素，它可被看作是一个子图像函数。如果 Z 表示实整数的集合，同时假设 (x,y) 是来自 $Z\times Z$ 的整数，f 和 b 是对坐标为 (x,y) 像素灰度值的函数（来自实数集 R 的实数）。如果灰度也是整数，则 Z 可由整数 R 所代替。

9.4.1 膨胀

函数 b 对函数 f 进行灰度膨胀可定义 $f\oplus b$，运算式如下：

$$(f\oplus b)(s,t)=\max\{f(s-x,t-y)+b(x,y)\mid(s-x),(t-y)\in D_f;(x,y)\in D_b\} \tag{9-49}$$

其中 D_f 和 D_b 分别是函数 f 和 b 的定义域，和前面一样，b 是形态处理的结构元素，不过此处 b 是一个函数而不是一个集合。

位移参数 $(s-x)$ 和 $(t-y)$ 必须包含在函数 f 的定义域内，此时它模仿二值膨胀运算定义。在这里两个集合必须至少有一个元素相交叠。还可以注意到，公式(9-48)非常类似于二维卷积公式，同时，在这里用“最大”代替卷积求和并以“相加”代替相乘。

下面我们将用一维函数来解释公式(9-49)中的运算原理。对于仅有一个变量的函数，公式(9-49)可以简化为

$$(f\oplus b)(s)=\max\{f(s-x)+b(x)\mid(s-x)\in D_f;x\in D_b\} \tag{9-50}$$

在卷积中，$f(-x)$ 仅是 $f(x)$ 关于 x 轴原点的映射，正像卷积运算那样，相对于正的 s，函数 $f(s-x)$ 将向右移，对于 $-s$，函数 $f(s-x)$ 将向左移。其条件是 $(s-x)$ 必须在 f 的定义域内，x 的值必须在 b 的定义域内。这意味着 f 和 b 将相覆盖，即 b 应包含在 f 内。这和二值图像膨胀定义要求的情形是类似的，即俩个集合至少应有一个元素是相互覆盖的。最后，与二值图像的情况不同，不是结构元素 b 而是 f 平移。公式(9-49)可以使 b 代替 f 写成平移的形式。然而，如果 D_b 比 D_f 小（这是实际中常见的），公式(9-49)所给出的形式就可在索引项中加以简化，并可以获得同样的结果。就概念而言，在 f 上滑动 b 和在 b 上滑动 f 是没有区别的。

膨胀是可以代换的，因而 f 和 b 相互代换的方法运用于公式(9-49)可以用来计算，$b\oplus f$ 结果都是一样的，而且 b 是平移函数。相反，腐蚀是不可交换的，因而，这种函数也是不可互换的。膨胀的例子可参见图 9-19。

由于膨胀操作是由结构元素形状定义的邻域中选择 $f-b$ 的最大值，因而通常对灰度图像的膨胀处理方法可得到两种结果：(1)如果所有的结构元素都为正，则输出图像将趋向比输入图像亮；(2)黑色细节减少或去除取决于在膨胀操作中结构元素相关的值和形状。

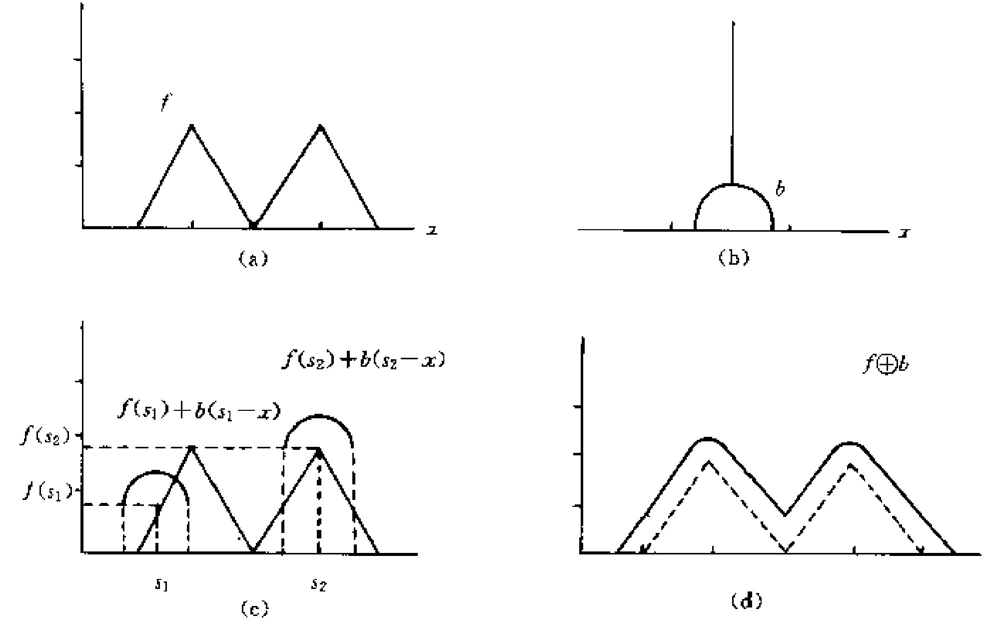


图 9-19 灰度膨胀图例

9.4.2 腐蚀

灰度图像的腐蚀定义为 $f\ominus b$，其运算公式为

$$(f\ominus b)(s,t)=\min\{f(s+x,t+y)-b(x,y)\mid(s+x),(t+y)\in D_f;(x,y)\in D_b\} \tag{9-51}$$

公式中 D_f 和 D_b 分别是 f 和 b 的定义域。平移参数 $(s+x)$ 和 $(t+y)$ 必须包含在 f 的定义域内，与二元腐蚀的定义类似，所有的结构元素将完全包含在与被腐蚀的集合内。还应注意到公式 (9-51)的形式与二维相关公式相似，只是用“最小”取代求和，用减法代替乘积。

如果只有一个变量时，我们可以用一维的腐蚀来说明公式(9-51)的原理。此时，表达式可简化为

$$(f\ominus b)(s)=\min\{f(s+x)-b(x)\mid(s-x)\in D_f;(x)\in D_b\} \tag{9-52}$$

在相关情况下，当 s 为正时，函数 $f(s+x)$ 将向右平移，当 s 为负时，函数 $f(s+x)$ 将移向左边，同时，要求 $(s-x)\in D_f$，$x\in D_b$ 意味着 b 将包含在 f 的范围内。这一点同二值图像腐蚀定义的情况相似，所有的结构元素将完全包含在被腐蚀的集合内。

不同于二值图像腐蚀定义，操作中是 f 在平移，而不是结构元素 b 在平移。公式(9-51)可以把 b 写成平移函数，由于 f 在 b 上滑动同 b 在 f 上滑动在概念上是一致的。图 9-20 展示了通过图 9-20(b)的结构元素腐蚀图 9-20(a)函数的结果。

正如公式(9-51)所示，腐蚀是在结构元素定义的领域内选择 $(f-b)$ 的最小值，因而，通常对灰度图像的膨胀处理可得到两种结果：(1)如果所有的结构元素都为正，则输出图像将趋向比输入图像暗；(2)在比结构元素还小的区域中的明亮细节经腐蚀处理后其效果将减弱。减弱的程度取决于环绕亮度区域的灰度值以及结构元素自身的形状和幅值。

与求补、映射相关的膨胀、腐蚀是有互补性的，即

$$(f\ominus b)^c(x,y)=(f^c\oplus\hat{b})(x,y) \tag{9-53}$$

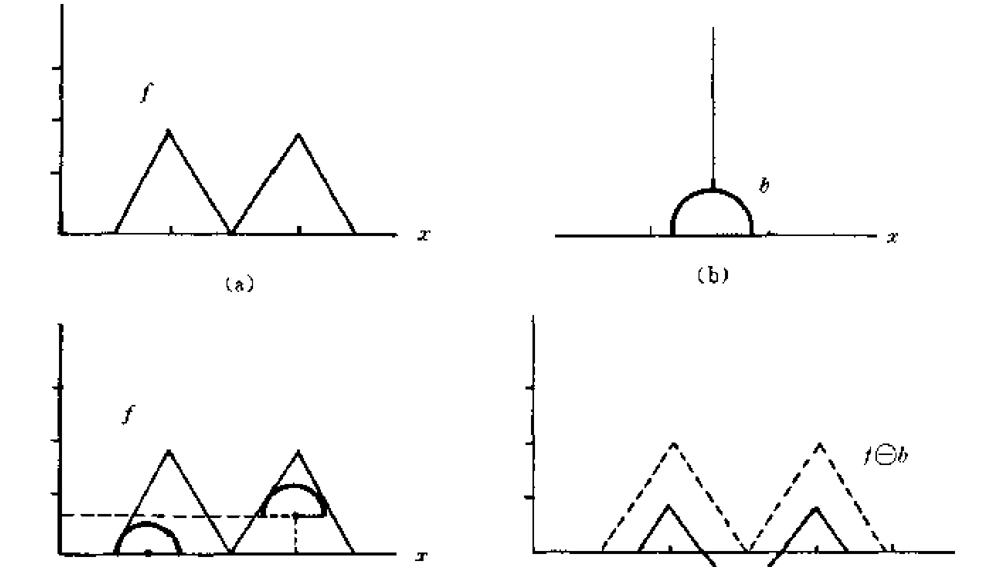


图 9-20 灰度腐蚀图例

其中 $f^c=-f(x,y)$；$\hat{b}=(-x,-y)$。

9.4.3 开和闭运算

灰度图像开运算和闭运算的表达式与二值图像相比具有相同的形式。结构元素 b 对图像 f 作开运算处理，可定义为 $f\circ b$，即

$$f\circ b=(f\ominus b)\oplus b \tag{9-54}$$

如果是二值图像的情况，开运算是 b 对 f 的简单的腐蚀操作，接下来对腐蚀的结果再进行膨胀操作。类似地，b 对 f 的闭运算，定义为 $f\bullet b$，即

$$f\bullet b=(f\oplus b)\circ b \tag{9-55}$$

灰度图像开运算和闭运算对于求补和映射也是对偶的，即

$$(f\bullet b)^c=f^c\circ\hat{b} \tag{9-56}$$

由于 $f^c=-f(x,y)$，式 (9-56) 也可以写为 $-(f\bullet b)=(-f\circ b)$。

图像的开和闭运算有一个简单的几何解释。

假设看到一个二维的图像函数 $f(x,y)$（像一个地貌地图），x 和 y 是空间坐标轴、第三坐标轴是亮度坐标轴（即：f 的值）。在重现中，图像作为一个平面显示，其中的任意点 (x,y) 是 f 在该点坐标值。假设我们想用球形结构元素 b 对 f 作开运算，这时可将 b 看作"滚动的球"。b 对 f 的开运算处理在几何上可解释为让"滚动球"沿 f 的下沿滚动，经这一"滚动"处理，所有的比"小球"直径小的峰都磨平了。图 9-21 解释了这一概念。图 9-21(a) 为解释简单，把灰度图像简化为连续函数剖面线；(b)显示了"滚动球"在不同的位置上滚动；(c)显示了沿函数剖面线结构元素 b 对 f 开运算处理的结果。所有小于球体直径的波峰值、尖锐度都减小了。在实际运用中，开运算处理常用于去除较小的亮点（相对结构元素而言），同时保留所有的灰度和较大的亮区特征不变。腐蚀操作去除较小的亮的细节，同时使图像变暗。如果再施以膨胀处理将增加图像的亮度而不再引入已去除的部分。

图 9-21(d)显示了结构元素 b 对 f 的闭操作处理。此时，小球(结构元素)在函数剖面上沿滚动，图 9-21(e)给出了处理结果，只要波峰的最窄部分超过小球的直径则波峰保留原来的形状。在实际运用中，闭运算处理常用于去除图像中较小的暗点(较结构元素而言)，同时保留原来较大的亮度特征。最初的膨胀运算去除较小暗细节，同时也使图像增亮。随后的腐蚀运算将图像调暗而不重新引入已去除的部分。

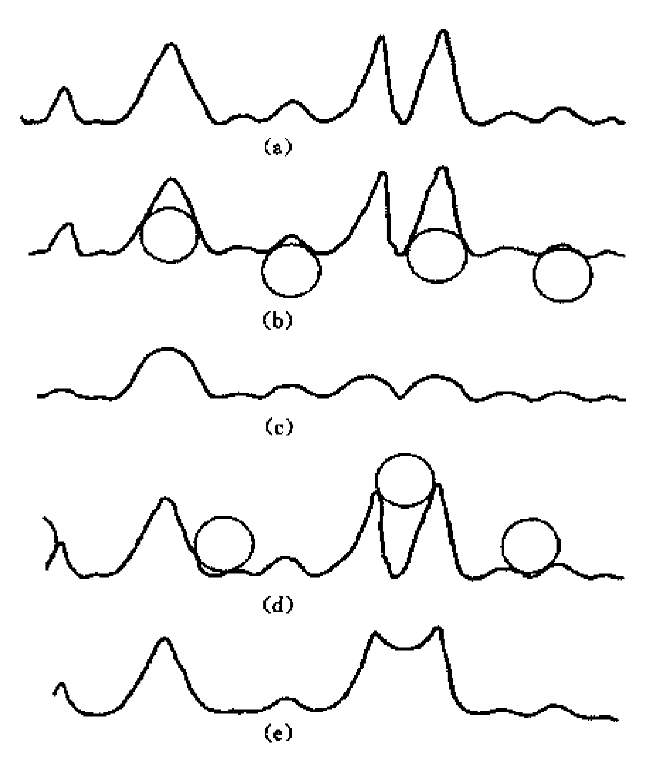


图 9-21　开和闭运算的图例

开运算处理满足以下的性质：

(i) $(f \circ b) \dashv f$；

(ii) 如果 $f_1 \dashv f_2$，则$(f_1 \circ b) \dashv (f_2 \circ b)$；

(iii) $(f \circ b) \circ b = f \circ b$。

表达式 $u \dashv v$ 表示 u 是 v 的子集，而且在 u 的定义域内对于任意(x,y)都有 $u(x,y) \leqslant v(x,y)$。

类似地，闭运算处理满足以下的性质：

(i) $f \dashv (f \bullet b)$；

(ii) 如果 $f_1 \dashv f_2$，则$(f_1 \bullet b) \dashv (f_2 \bullet b)$；

(iii) $(f \bullet b) \bullet b = f \bullet b$。

这些表达式的使用类似于对应的二值表达式。正如在二值情况下，对开运算处理和闭运算处理性质(ii)和性质(iii)被分别称作单调增加和等幂。

9.4.4 灰度形态学的应用

根据前边讨论的灰度形态学的基本运算，下边介绍一些简单的形态学实用处理算法，这些处理都是针对灰度图像进行的。

(1) 形态学图像平滑

一种获得平滑的方法是将图像先进行闭运算处理然后再进行开运算处理，处理结果将去除或消减亮斑和暗斑。

(2) 形态学图像梯度

除了前面对去除亮点和暗斑处理外，膨胀和腐蚀处理常用于计算图像的形态梯度，梯度用 g 表示，则

$$g=(f\oplus b)-(f\ominus b) \tag{9-57}$$

经过形态学梯度处理，使输入图像灰度变化更加尖锐，与利用像 Sobel 算子这样的一类处理方法所获得的梯度图像相反，运用对称结构元素获得的形态学梯度将较少受边缘方向的影响，这一优点的获得是以运算量显著增加为代价的。

(3) Top-hat 变换

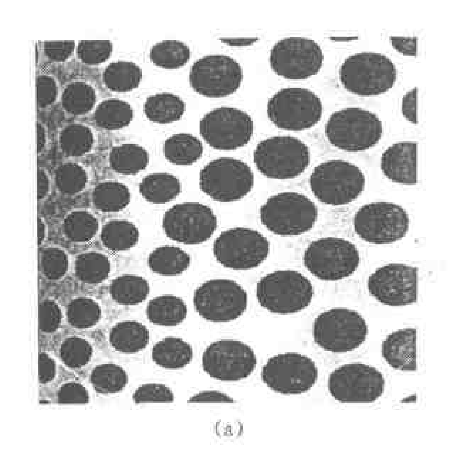
所谓的图像形态 Top-hat 变换用 h 来表示，其定义为

$$h=f-(f\circ b) \tag{9-58}$$

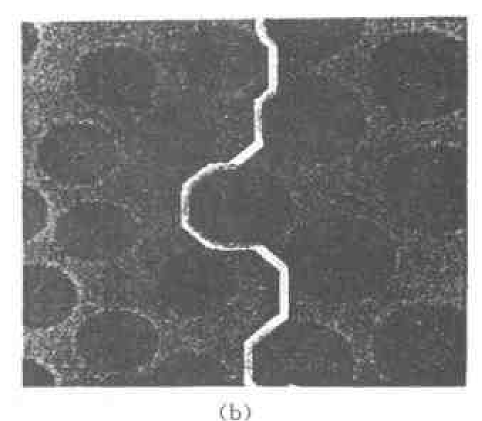
公式中 f 是输入图像，b 是结构元素函数。这一变换的最初命名是由于用平顶圆柱和平行六面体作为结构元素函数，因此，得名 Top-hat(高帽)变换，它常被用于阴影的细节增强处理。

(4) 纹理分割

图 9-22(a)是一幅包含两个纹理区的图像。我们的目的是分割出两个纹理区并提取两个区域的边界。由于闭运算可去除图像中的暗细节，在这种特殊情况下，依次使用较大的结构元素对输入图像进行闭运算处理。当结构元素的尺寸与小圆的尺寸相当时，它们将从图像中被除去，在原来的位置仅留下小圆曾经占有的区域的亮的背景。处理到这种状态，仅有右边大圆区域和左边背景区域。下一步，采用相对于大圆间的间隙来说较大的结构元素作开运算处理，将去除圆间的亮的区域，同时仅留下右边包含大圆的暗区域，这样，处理的结果将产生一个右边为暗，左边为亮的区域。用一个简单的门限就可以检测出两个区域。



(a)



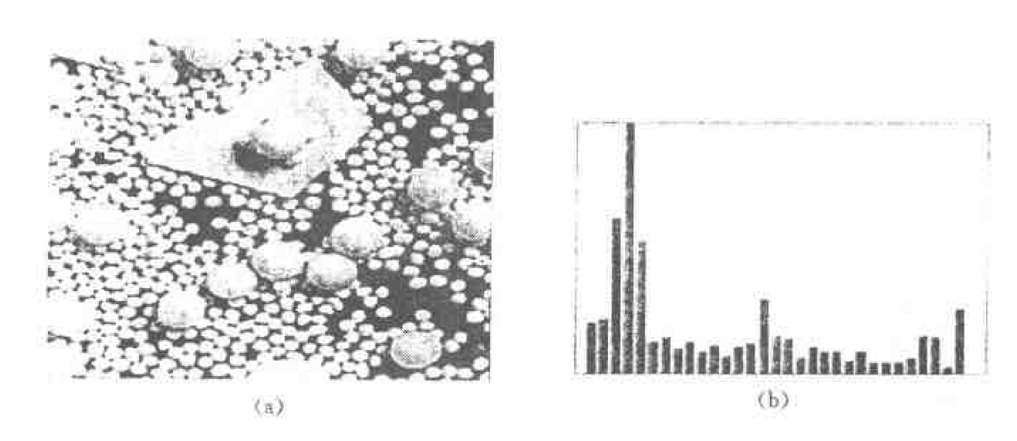
(b)

图 9-22　纹理分割

(5) 粒状处理

粒状处理和其他处理一样，是决定一幅图像分散颗粒尺寸大小的处理。图 9-23(a)显示了包含三种不同尺寸的亮颗粒图像。这些颗粒不但重叠，而且混乱到无法检测单一个体的程度。由于与背景相比颗粒较亮，形态学处理将可以用来决定尺寸的分布。首先对原始图像用不断增大尺寸的结构元素进行开运算处理。经过不同的结构元素的开处理后，原始图像和开运算处理后图像的差异可以被算出，处理最后，这些区别将先被归一化并作出颗粒分布的直方图。这一

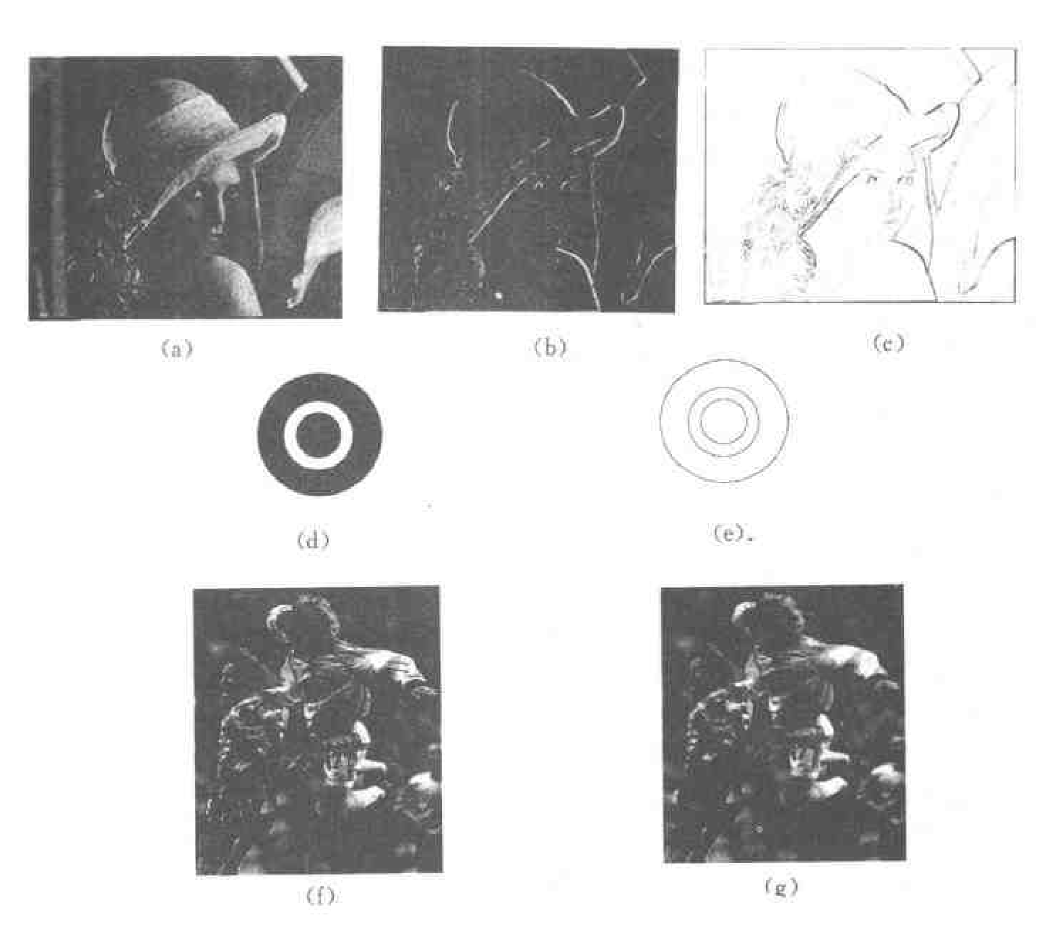
方法是基于这样的观点：对特殊尺寸的开运算处理将对包含最小尺寸颗粒的输入图像最有效。因此通过计算输入图像和输出图像的差异将可获得对这些颗粒的相对数量的测量。图 9-23(b)显示了这种情况的结果。直方图表明了输入图像中最多的三种颗粒分布。



(a) (b)

图 9-23 颗粒图像处理

图 9-24 示出了数学形态学基本处理的结果。同时，在附录 9 中给出了数学形态学的基本处理程序，供读者参考。



(a) (b) (c) (d) (e) (f) (g)

图 9-24 形态学处理效果

(a)原图，(b)梯度处理结果，(c)边缘提取结果，(d)原图二值图像，

(e)二值边缘提取处理结果，(f)原图像，(g) 平滑处理结果

附录 9 数学形态学处理程序实例

```
/ JEFFView.cpp : implementation of the CJEFFView class
//

#include "stdafx.h"
#include "JEFF.h"

#include "JEFFDoc.h"
#include "JEFFView.h"
#define Out(x,y) lpPoints[(x)+(y)*nWidth]
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[]=_FILE_;
#endif
//////////////////////////////////////////
// CJEFFView
IMPLEMENT_DYNCREATE(CJEFFView, CScrollView)
BEGIN_MESSAGE_MAP(CJEFFView, CScrollView)
    //{{AFX_MSG_MAP(CJEFFView)
    ON_COMMAND(ID_EROSION, OnErosion)
    ON_COMMAND(ID_DILATION, OnDilation)
    ON_COMMAND(ID_OPEN, OnOpen)
    ON_COMMAND(ID_CLOSE, OnClose)
    ON_COMMAND(ID_ZAOSHENG, OnZaosheng)
    ON_COMMAND(ID_BIANYUAN, OnBianyuan)
    ON_COMMAND(ID_1, On1)
    ON_COMMAND(ID_2, On2)
    ON_COMMAND(ID_4, On4)
    ON_COMMAND(ID_8, On8)
    ON_COMMAND(ID_DILATION2, OnDilation2)
    ON_COMMAND(ID_EROSION2, OnErosion2)
    ON_COMMAND(ID_OPEN2, OnOpen2)
    ON_COMMAND(ID_CLOSE2, OnClose2)
    ON_COMMAND(ID_SMOOTH, OnSmooth)
    ON_COMMAND(ID_RELOAD, OnReload)
    ON_COMMAND(ID_TIDU, OnTidu)
    ON_COMMAND(ID_BIANYUAN_H, OnBianyuanH)
    ON_COMMAND(ID_QUSE, OnQuse)
    ON_WM_LBUTTONDOWN()
    //}}AFX_MSG_MAP
    // Standard printing commands
```

```
    ON_COMMAND(ID_FILE_PRINT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, CScrollView::OnFilePrintPreview)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CJEFFView construction/destruction

CJEFFView::CJEFFView()
{
    // TODO: add construction code here
    ValidDoc=FALSE;
    m_i=1;
}

CJEFFView::~CJEFFView()
{
}

BOOL CJEFFView::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CScrollView::PreCreateWindow(cs);
}

/////////////////////////////////////////////////////////////////////////////
// CJEFFView drawing

void CJEFFView::OnDraw(CDC * pDC)
{
    CJEFFDoc * pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    // TODO: add draw code for native data here
    if (pDoc->lpBitmap)
    {
        ValidDoc=TRUE;
        nWidth=pDoc->nWidth;
        nHeight=pDoc->nHeight;
        lpBits=pDoc->lpBits;
        lpPoints=pDoc->lpPoints;
```

```
        BITMAPINFOHEADER bmi;
        bmi.biSize=sizeof(BITMAPINFOHEADER);
        bmi.biWidth=nWidth;
        bmi.biHeight= nHeight;
        bmi.biPlanes=1;
        bmi.biBitCount=24;
        bmi.biCompression=BI_RGB;
        bmi.biSizeImage=0;
        bmi.biXPelsPerMeter=0;
        bmi.biYPelsPerMeter=0;
        bmi.biClrUsed=0;
        bmi.biClrImportant=0;
        StretchDIBits(pDC->m_hDC,0,0,nWidth*m_i,nHeight*m_i,0,0,nWidth,nHeight,
        lpBits,
        (BITMAPINFO *)&bmi,
        DIB_RGB_COLORS,
        SRCCOPY);
    }
}

void CJEFFView::OnInitialUpdate()
{
    CScrollView::OnInitialUpdate();
    CSize sizeTotal;
    // TODO: calculate the total size of this view
    sizeTotal.cx = 2000; sizeTotal.cy = 1700;
    SetScrollSizes(MM_TEXT, sizeTotal);
}

/////////////////////////////////////////////////////////////////////////////
// CJEFFView printing

BOOL CJEFFView::OnPreparePrinting(CPrintInfo* pInfo)
{
    // default preparation
    return DoPreparePrinting(pInfo);
}

void CJEFFView::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add extra initialization before printing
}

void CJEFFView::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
```

```
{
    // TODO: add cleanup after printing
}

/////////////////////////////////////////////////////////////////////////////
// CJEFFView diagnostics

#ifdef _DEBUG
void CJEFFView::AssertValid() const
{
    CScrollView::AssertValid();
}

void CJEFFView::Dump(CDumpContext& dc) const
{
    CScrollView::Dump(dc);
}

CJEFFDoc * CJEFFView::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CJEFFDoc)));
    return (CJEFFDoc *)m_pDocument;
}
#endif //_DEBUG

/////////////////////////////////////////////////////////////////////////////
// CJEFFView message handlers

//二值形态学:腐蚀
void CJEFFView::OnErosion()
{
    int *lpMarks=new int[nWidth * nHeight];
    int x,y,x1,y1;
    int count;//中间变量
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
            count=1;
            for(y1=-1;y1<2;y1++)
            {
                for(x1=-1;x1<2;x1++)
                {
                  if ((Out(x+x1,y+y1))//以(x,y)为中心的9个点是否有白色的像素点
```

```
                    count=0;//如果有 count=0,否则 count=1;
                }
            }
            if (count) lpMarks[x+y * nWidth]=1;//黑色像素点记为 1
            else lpMarks[x+y * nWidth]=0;//白色像素点记为 0
        }
    }
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
            if (lpMarks[x+y * nWidth]==0) Out(x,y)=255;//白色像素点灰度为 255
        }
    }
    delete lpMarks;//释放内存
    GetDocument()->PutPoint();
    Invalidate();
}

//二值形态学:膨胀
void CJEFFView::OnDilation()
{
    int x,y,x1,y1;
    int count;//中间变量
    int * lpMarks=new int[nWidth * nHeight];
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
            count=0;
            for(y1=-1;y1<2;y1++)
            {
                for(x1=-1;x1<2;x1++)
                {
                    if (Out(x+x1,y+y1)==0)// 以(x,y)为中心的 9 个点是否有黑色的像素点
                        count=1;//若有记为一
                }
            }
            if (count) lpMarks[x+y * nWidth]=1;//黑色像素点记为 1
            else lpMarks[x+y * nWidth]=0;//白色像素点记为 0
        }
    }
    for(y=1;y<nHeight-1;y++)
    {
```

```
        for(x=1;x<nWidth-1;x++)
        {
            if (lpMarks[x+y*nWidth]) Out(x,y)=0;//黑色像素点输出灰度值为0
        }
    }
delete lpMarks;
    GetDocument()->PutPoint();
    Invalidate();
}

//二值形态学:开启
void CJEFFView::OnOpen()
{
    //先腐蚀后膨胀
    OnErosion();
    OnDilation();
}

//二值形态学:闭合
void CJEFFView::OnClose()
{
    //先膨胀后腐蚀
    OnDilation();
    OnErosion();
}

//二值形态学:噪声滤除
void CJEFFView::OnZaosheng()
{
    //先开启后闭合
    OnOpen();
    OnClose();
}

//二值形态学:边缘提取
void CJEFFView::OnBianyuan()
{
    int *lpMarks=new int[nWidth*nHeight];//用于保存腐蚀图像灰度值
    int *lpMarks1=new int[nWidth*nHeight]; //用于保存原图像灰度值
    int x,y,x1,y1,count;
    for(y=0;y<nHeight-1;y++)
    {
        for(x=0;x<nWidth-1;x++)
        {
```

```
            lpMarks1[x+y * nWidth]=Out(x,y);//原图像保存至 lpMarks1
        }
    }
    //以下为腐蚀操作
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
            count=1;
            for(y1=-1;y1<2;y1++)
            {
                for(x1=-1;x1<2;x1++)
                {
                    if (Out(x+x1,y+y1))
                        count=0;
                }
            }
            if (count) lpMarks[x+y * nWidth]=0;
            else lpMarks[x+y * nWidth]=255;
        }
    }
    //取原图像与腐蚀图像的差集为输出
    for(y=0;y<nHeight-1;y++)
    {
        for(x=0;x<nWidth-1;x++)
        {
            if(lpMarks1[x+y * nWidth]==lpMarks[x+y * nWidth]) Out(x,y)=255;
            else Out(x,y)=0;
        }
    }
    delete lpMarks;//释放内存
    delete lpMarks1;//释放内存
    GetDocument()->PutPoint();
    Invalidate();
}

//放大镜:1 倍
void CJEFFView::On1()
{
    m_r=1;
    Invalidate();
}

//放大镜:2 倍
```

```
void CJEFFView::On2()
{
    m_i=2;
    Invalidate();
}

//放大镜:4 倍
void CJEFFView::On4()
{
    m_i=4;
    Invalidate();
}

//放大镜:8 倍
void CJEFFView::On8()
{
    m_i=8;
    Invalidate();
}

//灰度形态学:膨胀
void CJEFFView::OnDilation2()
{
    int *lpMarks=new int[nWidth*nHeight];//定义一块内存,大小为图像像素数
    int x,y,x1,y1,s,t;
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
          s=t=255;//s,t 为中间变量,因为要找最小值,所以定义其为 255
          for(y1=-1;y1<2;y1++)
          {
              for(x1=-1;x1<2;x1++)
              {
              s=Out(x-x1,y+y1);//以点(x,y)为中心的 9 个点进行操作
              if(s<t) t=s;//将 9 个点灰度的最小值传给 t
              }
          }
              lpMarks[x+y*nWidth]=t;//将处理后的图像灰度值暂存于内存中
        }
    }
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
```

```
        {
          Out(x,y)=lpMarks[x+y * nWidth];//传至输出
        }
    }
    delete lpMarks;//释放内存
    GetDocument()->PutPoint();
    Invalidate();
}

//灰度形态学:腐蚀
void CJEFFView::OnErosion2()
{
    int x,y,x1,y1,s,t;
    int * lpMarks=new int[nWidth * nHeight];//定义一块内存,大小为图像像素数
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
        s=t=0;//s,t 为中间变量,因为要找最大值,所以定义其为 0
          for(y1=-1;y1<2;y1++)
          {
            for(x1=-1;x1<2;x1++)
            {
              s=Out(x+x1,y+y1);//以点(x,y)为中心的 9 个点进行操作
              if(s>t) t=s;//将 9 个点灰度的最大值传给 t
            }
        }
        lpMarks[x+y * nWidth]=t;//将处理后的图像灰度值暂存于内存中
        }
    }
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
          Out(x,y)=lpMarks[x+y * nWidth];//传至输出
        }
    }
    delete lpMarks;//释放内存
    GetDocument()->PutPoint();
    Invalidate();
}

//灰度形态学:开启
void CJEFFView::OnOpen2()
```

```
{
    //先腐蚀后膨胀
    OnErosion2();
    OnDilation2();
}

//灰度形态学:闭合
void CJEFFView::OnClose2()
{
    //先膨胀后腐蚀
    OnDilation2();
    OnErosion2();
}

//灰度形态学:平滑
void CJEFFView::OnSmooth()
{
    //先开启后闭合
    OnOpen2();
    OnClose2();
}

//灰度形态学:梯度
void CJEFFView::OnTidu()
{
    // TODO: Add your command handler code here
    int *lpMarks=new int[nWidth * nHeight];
    int x,y,x1,y1,s,t;
    for(y=1;y<nHeight-1;y++)
    {
         for(x=1;x<nWidth-1;x++)
         {
           s=t=255;
           for(y1=-1;y1<2;y1++)
           {
               for(x1=-1;x1<2;x1++)
               {
               s=Out(x+x1,y+y1);
             if(s<t) t=s;
           }
       }
       lpMarks[x+y*nWidth]=t;

    }
```

```
    int * lpMarks1=new int[nWidth * nHeight];
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
            s=t=0;
            for(y1=-1;y1<2;y1++)
            {
                for(x1=-1;x1<2;x1++)
                {
                    s=Out(x+x1,y+y1);
                    if(s>t) t=s;
                }
            }
            lpMarks1[x+y * nWidth]=t;
        }
    }
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
            if (lpMarks[x+y * nWidth]-lpMarks1[x+y * nWidth]==0) Out(x,y)=255;
            else Out(x,y)=lpMarks[x+y * nWidth]-lpMarks1[x+y * nWidth];

        }
    }
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {

            Out(x,y)=255-Out(x,y);

        }
    }
    delete lpMarks;
    delete lpMarks1;
    GetDocument()->PutPoint();
    Invalidate();
}

//重载图片
void CJEFFView::OnReload()
{
```

```
    // TODO: Add your command handler code here
    CJEFFDoc * pDoc = GetDocument();
    CString path = pDoc->GetPathName();
    pDoc->OnNewDocument();
    pDoc->OnOpenDocument(path);
    pDoc->SetPathName(path);
    Invalidate();


//灰度形态学:边缘提取
void CJEFFView::OnBianyuanH()
{
    int * lpMarks = new int[nWidth * nHeight]; //用于保存腐蚀图像
    int * lpMarks1 = new int[nWidth * nHeight]; //用于保存原图像
    int x,y,x1,y1,s,t;
    for(y=0;y<nHeight-1;y++)
    {
        for(x=0;x<nWidth-1;x++)
        {
            lpMarks1[x+y*nWidth]=Out(x,y); //将原始图片灰度值存于lpMarks1中
        }
    }
    //以下为腐蚀操作
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
        {
          s=t=0;
          for(y1=-1;y1<2;y1++)
          {
              for(x1=-1;x1<2;x1++)
              {
                  s=Out(x-x1,y-y1);
                  if(s>t) t=s;
              }
          }
          lpMarks[x+y*nWidth]=t; //腐蚀操作结果存于lpMarks中
        }
    }
    //用原始图像灰度值减去腐蚀操作结果灰度值
    for(y=1;y<nHeight-1;y++)
    {
        for(x=1;x<nWidth-1;x++)
```

```
            if (lpMarks1[x+y * nWidth]-lpMarks[x+y * nWidth]<=0) Out(x,y)=255;
            else Out(x,y)=lpMarks1[x+y * nWidth]-lpMarks[x+y * nWidth];

        }
    }
    delete lpMarks;//释放内存
    delete lpMarks1;//释放内存
    GetDocument()->PutPoint();
    Invalidate();
}

void CJEFFView::OnQuse()
{
    GetDocument()->PutPoint();
    Invalidate();
}

void CJEFFView::OnLButtonDown(UINT nFlags, CPoint point)
{
     CClientDC dc(this);
     char str[32];
     sprintf(str,"(%d,%d)",point.x, point.y);
     dc.TextOut(0,0,str,strlen(str));
}
```

```
// JEFFView.h : interface of the CJEFFView class
//
/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX_JEFFVIEW_H__E252A5AE_1DAA_11D4_9ABC_89D139F8CC3D__INCLUDED_)
#define AFX_JEFFVIEW_H__E252A5AE_1DAA_11D4_9ABC_89D139F8CC3D__INCLUDED_

#if _MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000

class CJEFFView : public CScrollView
{
protected: // create from serialization only
    CJEFFView();
    DECLARE_DYNCREATE(CJEFFView)

// Attributes
```

```
public:
    CJEFFDoc * GetDocument();
    BOOL ValidDoc;
    int nWidth;
    int nHeight;
    BYTE  * lpBits;
    BYTE  * lpPoints;
// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CJEFFView)
    public:
    virtual void OnDraw(CDC *  pDC); // overridden to draw this view
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    protected:
    virtual void OnInitialUpdate(); // called first time after construct
    virtual BOOL OnPreparePrinting(CPrintInfo *  pInfo);
    virtual void OnBeginPrinting(CDC *  pDC, CPrintInfo *  pInfo);
    virtual void OnEndPrinting(CDC *  pDC, CPrintInfo *  pInfo);
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual~CJEFFView();
    int m_i;
#ifdef  DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
    //{{AFX_MSG(CJEFFView)
    afx_msg void OnErosion();
    afx_msg void OnDilation();
    afx_msg void OnOpen();
    afx_msg void OnClose();
    afx_msg void OnZaosheng();
    afx_msg void OnBianyuan();
    afx_msg void On1();
```

```
    afx_msg void On2();
    afx_msg void On4();
    afx_msg void On8();
    afx_msg void OnDilation2();
    afx_msg void OnErosion2();
    afx_msg void OnOpen2();
    afx_msg void OnClose2();
    afx_msg void OnSmooth();
    afx_msg void OnReload();
    afx_msg void OnTidu();
    afx_msg void OnBianyuanH();
    afx_msg void OnQuse();
    afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

#ifndef _DEBUG // debug version in JEFFView.cpp
inline CJEFFDoc * CJEFFView::GetDocument()
   { return (CJEFFDoc *)m_pDocument; }
#endif

/////////////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Developer Studio will insert additional declarations immediately before the previous line.

#endif
// !defined(AFX_JEFFVIEW_H__E252A5AE_1DAA_11D1_9ABC_89D139F8CC3D__INCLUDED_)
```

```
// JEFFDoc.cpp : implementation of the CJEFFDoc class
//

#include "stdafx.h"
#include "JEFF.h"

#include "JEFFDoc.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
//////////////////////////////////////////////
// CJEFFDoc

IMPLEMENT_DYNCREATE(CJEFFDoc, CDocument)

BEGIN_MESSAGE_MAP(CJEFFDoc, CDocument)
    //{{AFX_MSG_MAP(CJEFFDoc)
        // NOTE-the ClassWizard will add and remove mapping macros here.
        //     DO NOT EDIT what you see in these blocks of generated code!
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

//////////////////////////////////////////////
// CJEFFDoc construction/destruction

CJEFFDoc::CJEFFDoc()
{
    // TODO: add one-time construction code here
    lpBitmap=0;
    lpPoints=0;
}

CJEFFDoc::~CJEFFDoc()
{
    if (lpBitmap) delete lpBitmap;
    if (lpPoints) delete lpPoints;
}

BOOL CJEFFDoc::OnNewDocument()
{
    if (!CDocument::OnNewDocument())
        return FALSE;

    // TODO: add reinitialization code here
    // (SDI documents will reuse this document)

    return TRUE;
}

//////////////////////////////////////////////
// CJEFFDoc serialization

void CJEFFDoc::Serialize(CArchive& ar)
```

```
CFile *pFile;
BITMAPINFOHEADER *pInfo;
if (ar.IsStoring())
{
    // TODO: add storing code here
    if (lpBitmap)
    {
        ar.Write(lpBitmap,nLen);
    }
}
else
{
    // TODO: add loading code here
    if (lpBitmap) delete lpBitmap;
    if (lpPoints) delete lpPoints;
    ar.Flush();
    pFile=ar.GetFile();
    nLen=pFile->GetLength();
    lpBitmap=new BYTE[nLen];
    ar.Read(lpBitmap,nLen);
    //get file type
    pInfo=(BITMAPINFOHEADER *)(lpBitmap-sizeof(BITMAPFILEHEADER));
    nWidth=pInfo->biWidth;
    nByteWidth=nWidth * 3;
    if (nByteWidth%4) nByteWidth+=4-(nByteWidth%4);
    nHeight=pInfo->biHeight;
    if (pInfo->biBitCount!=24)
    {
        if (pInfo->biBitCount!=8)
        {
            AfxMessageBox("无效位图");
            delete lpBitmap;
            lpBitmap=0;
            return;

        unsigned int PaletteSize=1<<pInfo->biBitCount;
        if (pInfo->biClrUsed!=0 && pInfo->biClrUsed<PaletteSize) PaletteSize=pInfo->
        biClrUsed;
        lpBits=lpBitmap+sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
        RGBQUAD *pPalette=(RGBQUAD *)lpBits;
        lpBits+=sizeof(RGBQUAD) * PaletteSize;
nLen  sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER)-nByteWidth * nHeight;
        BYTE *lpTemp  lpBitmap;
        lpBitmap=new BYTE[nLen];
```

```
BITMAPFILEHEADER bmh;
BITMAPINFOHEADER
    bmi;
bmh.bfType='B'+'M'*256;
bmh.bfSize=nLen;
bmh.bfReserved1=0;
bmh.bfReserved2=0;
bmh.bfOffBits=54;

bmi.biSize=sizeof(BITMAPINFOHEADER);
bmi.biWidth=nWidth;
bmi.biHeight=nHeight;
bmi.biPlanes=1;
bmi.biBitCount=24;
bmi.biCompression=BI_RGB;
bmi.biSizeImage=0;
bmi.biXPelsPerMeter=0;
bmi.biYPelsPerMeter=0;
bmi.biClrUsed=0;
bmi.biClrImportant=0;
int nBWidth=pInfo->biWidth;
if (nBWidth%4) nBWidth+=4-(nBWidth%4);
memset(lpBitmap,0,nLen);
memcpy(lpBitmap,&bmh,sizeof(BITMAPFILEHEADER));
memcpy(lpBitmap+sizeof(BITMAPFILEHEADER),&bmi,sizeof(BITMAPINFOHEADER));
BYTE
*lpBits2=lpBitmap+sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
int x,y,p1,p2,Palette;
for(y=0;y<nHeight;y++)
{
    for(x=0;x<nWidth;x++)
    {
        p1=y*nBWidth+x;
        p2=y*nByteWidth+x*3;
        if (lpBits[p1]<PaletteSize) Palette=lpBits[p1];
        else Palette=0;
        lpBits2[p2]=pPalette[Palette].rgbBlue;
        lpBits2[p2+1]=pPalette[Palette].rgbGreen;
        lpBits2[p2+2]=pPalette[Palette].rgbRed;
    }
}
delete lpTemp;
```

```
        lpBits =lpBitmap+sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
        lpPoints=new unsigned char[nWidth * nHeight];
        GetPoint();
    }
}

/////////////////////////////////////////////////////////////////////////////
// CJEFFDoc diagnostics

#ifdef _DEBUG
void CJEFFDoc::AssertValid() const
{
    CDocument::AssertValid();
}

void CJEFFDoc::Dump(CDumpContext& dc) const
{
    CDocument::Dump(dc);
}
#endif //_DEBUG

/////////////////////////////////////////////////////////////////////////////
// CJEFFDoc commands

void CJEFFDoc::GetPoint()
{
    int x,y,p;
    BOOL SetGray=TRUE;
    for(y=0;y<nHeight;y++)
    {
        for(x=0;x<nWidth;x++)
        {
            p=x*3+(nHeight-y-1)*nByteWidth;
            lpPoints[x+y*nWidth]=(BYTE)(0.2125*(double)lpBits[p+2]+0.7154*(double)lpBits[p
+1]+0.0721*(double)lpBits[p]);
            if (lpBits[p+2]!=lpBits[p+1] || lpBits[p+1]!=lpBits[p]) SetGray=FALSE;
        }
    }
    if (SetGray) Gray=TRUE;
}

void CJEFFDoc::PutPoint()
{
    int x,y,p,p1;
```

```
    for(y=0;y<nHeight;y++)
    {
        for(x=0;x<nWidth;x++)
        {
            p=x*3+(nHeight-y-1)*nByteWidth;
            p1=x+y*nWidth;
            lpBits[p]=lpPoints[p1];
            lpBits[p+1]=lpPoints[p1];
            lpBits[p+2]=lpPoints[p1];
        }
    }
    Gray=TRUE;
```

```
// JEFFDoc.h : interface of the CJEFFDoc class
//
/////////////////////////////////////////////////////////////////////////////

#if !defined(AFX_JEFFDOC_H__E252A5AC_1DAA_11D4_9ABC_89D139F8CC3D__INCLUDED_)
#define AFX_JEFFDOC_H__E252A5AC_1DAA_11D4_9ABC_89D139F8CC3D__INCLUDED_

#if _MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000


class CJEFFDoc : public CDocument
{
protected: // create from serialization only
    CJEFFDoc();
    DECLARE_DYNCREATE(CJEFFDoc)

// Attributes
public:
    BYTE *lpBitmap;
    int nWidth;
    int nHeight;
    int nLen;
    BYTE *lpPoints;
    BYTE *lpBits;
    int nByteWidth;
    BOOL Gray;
// Operations
public:
```

```
// Overrides
	// ClassWizard generated virtual function overrides
	//{{AFX_VIRTUAL(CJEFFDoc)
	public:
	virtual BOOL OnNewDocument();
	virtual void Serialize(CArchive& ar);
	//}}AFX_VIRTUAL

// Implementation
public:
	void PutPoint();
	void GetPoint();
	virtual ~CJEFFDoc();
#ifdef _DEBUG
	virtual void AssertValid() const;
	virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
	//{{AFX_MSG(CJEFFDoc)
		// NOTE - the ClassWizard will add and remove member functions here.
		//    DO NOT EDIT what you see in these blocks of generated code !
	//}}AFX_MSG
	DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Developer Studio will insert additional declarations immediately before the previous line.

#endif
// !defined(AFX_JEFFDOC_H__E252A5AC_1DAA_11D1_9ABC_89D139F8CC3D__INCLUDED_)
```

```
// MainFrm.cpp : implementation of the CMainFrame class
//

#include "stdafx.h"
#include "JEFF.h"
```

```
#include "MainFrm.h"
#include "Splash.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[]= __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CMainFrame

IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
    //{{AFX_MSG_MAP(CMainFrame)
    ON_WM_CREATE()
    //ON_COMMAND(ID_SMOOTH, OnSmooth)
    //ON_COMMAND(ID_BIANYUAN2, OnBianyuan2)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

static UINT indicators[] =
{
    ID_SEPARATOR,           // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};

/////////////////////////////////////////////////////////////////////////////
// CMainFrame construction/destruction

CMainFrame::CMainFrame()
{
    // TODO: add member initialization code here



}

CMainFrame::~CMainFrame()
{
}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
```

```
{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndToolBar.Create(this) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;      // fail to create
    }

    if (!m_wndStatusBar.Create(this) ||
        !m_wndStatusBar.SetIndicators(indicators,
         sizeof(indicators)/sizeof(UINT)))
    {
        TRACE0("Failed to create status bar\n");
        return -1;      // fail to create
    }

    // TODO: Remove this if you don't want tool tips or a resizeable toolbar
    m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle() |
        CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC);

    // TODO: Delete these three lines if you don't want the toolbar to
    // be dockable
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);

    // CG: The following line was added by the Splash Screen component.
    CSplashWnd::ShowSplashScreen(this);
    return 0;
}

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CFrameWnd::PreCreateWindow(cs);
}

/////////////////////////////////////////////////////////////////////////////
// CMainFrame diagnostics
```

```
#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CFrameWnd::AssertValid();

void CMainFrame::Dump(CDumpContext& dc) const
{
    CFrameWnd::Dump(dc);
}

#endif // DEBUG

///////////////////////////////////////////////
// CMainFrame message handlers
```

```
// MainFrm.h : interface of the CMainFrame class
//
///////////////////////////////////////////////

#if !defined(AFX_MAINFRM_H_E252A5AA_1DAA_11D4_9ABC_89D139F8CC3D_INCLUDED_)
#define AFX_MAINFRM_H_E252A5AA_1DAA_11D4_9ABC_89D139F8CC3D_INCLUDED
#if _MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000
class CMainFrame : public CFrameWnd

protected: // create from serialization only
    CMainFrame();
    DECLARE_DYNCREATE(CMainFrame)
  Attributes
public:
// Operations
public:
// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CMainFrame)
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    //}}AFX_VIRTUAL
  Implementation
public:
```

```
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected: // control bar embedded members
    CStatusBar  m_wndStatusBar;
    CToolBar    m_wndToolBar;

// Generated message map functions
protected:
    //{{AFX_MSG(CMainFrame)
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    afx_msg void OnSmooth();
    afx_msg void OnBianyuan2();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Developer Studio will insert additional declarations immediately before the previous line.

#endif
// !defined(AFX_MAINFRM_H_E252A5AA_1DAA_11D4_9ABC_89D139F8CC3D_INCLUDED_)
```

思 考 题

1. 如果把膨胀和腐蚀通过向量来运算，则膨胀可用下式来实现，即：

$\boldsymbol{A} \oplus \boldsymbol{B} = \{x \mid x = a + b;\ a \in \boldsymbol{A}, b \in \boldsymbol{B}\}$ 试证明该式与

$\boldsymbol{A} \oplus \boldsymbol{B} = \{x \mid [(\hat{\boldsymbol{B}})_x \cap \boldsymbol{A} \neq \varnothing]\}$ 是等价的。式中 A、B 都是向量。

2. 试证明腐蚀 $\boldsymbol{A} \ominus \boldsymbol{B} = \{x \mid (\boldsymbol{B})_x \subseteq \boldsymbol{A}\}$ 与

$\boldsymbol{A} \ominus \boldsymbol{B} = \{x \mid (x + b)\} \in A$，对于 $b \in \boldsymbol{B}$ 也是等价的，$\boldsymbol{A}$，$\boldsymbol{B}$ 都是向量。

3. 划出半径为 $r/8$ 的圆形结构元素膨胀一个半径为 r 的圆形图像的示意图。

4. 划出半径为 $r/8$ 的圆形结构元素腐蚀一个半径为 r 的圆形图像的示意图。

5. 设计一个形态学算法，将 8 连通二值边界转化为 m 连通二值边界，可假定原边界的宽度为一个像素宽度。

6. 设计一个形态学平滑处理程序（利用开、闭运算）。

7. 利用公式 $g = (f \oplus b) - (f \ominus b)$ 设计一个形态学梯度处理程序。

8. 设计一个 Top-hat 变换程序，观察图像处理结果。

第 10 章 模式识别的理论和方法

10.1 概述

模式识别是随着计算机的发展而兴起的一门新的技术科学。自50年代末期开始来，至今已得到了迅速的发展和广泛的应用。谈到模式识别，对我们每个人来说，每时每刻都在进行着，例如：医生看病，要了解许多情况，最后判断患了什么病，这是一个识别过程；读书、看报也是识别过程，不会识别就看不懂。用计算机进行模式识别就是研究让计算机处理哪些信息和怎样处理这些信息。因此，它是信息处理中的又一个研究领域。例如，根据气象观测数据或气象卫星拍照的照片如何准确地预报天气；根据石油勘探的人工地震波如何提供储油的岩层结构；从遥感图片中如何区别出农作物、湖泊、森林、导弹基地等；在高能物理实验中怎样识别粒子径迹；在医疗诊断中如何从X射线照片中发现病灶；如何根据信函上的邮政编码自动分拣信件；在繁华的交通中心根据车辆的流量如何决定开放红灯或绿灯等诸如此类的问题都是模式识别研究的课题。这些课题看上去五花八门，名目繁多，但总起来看主要是研究分类问题。模式识别研究的对象基本上可概括为二类：一是有直觉形象的如图片、相片、图案、文字等等，一种是无直觉形象而只有数据或信号波形如语言、声音、心电脉冲、地震波等等。但对模式识别来说，无论是数据、信号还是平面图形和物体，都是除掉它们的物理内容找出它们的共性，把具有同一共性的归为一类，有另一种共性者则归为另一类。例如：10个阿拉伯数字分为10类；26个英文字母分成26类；白血球有5种就分为5类；肺部X射线照片可分为异常和正常两类等等。

模式识别研究的目的是构造自动处理某些信息的机器系统，以代替人完成分类和辨识的任务。特别是有直觉形象的一类图像识别问题同人或其他动物的感知活动尤其同人脑的智力活动联系密切。因此，根据人的大脑识别的机理，在工程上用计算机模拟，从而研究识别方法是有现实意义的。尽管这种模拟同人的意识和思维活动有本质的差别，但若从人类识别图像的过程及认识规律中得到启发，在某些环节上得到借鉴，从而采用现代技术解决实际问题这是十分有益的。在具有视觉形象的图像识别中，许多方法和概念就是从人类认识图像的过程中直接移植过来的。人类在现实生活中要区别各种现象、物体及声音，一般总是首先抓住它们的特征进行比较、分析、判断，从而将它们分类或识别。特别是数理统计和模糊数学的发展，总结了人们的认识逻辑，从而也使图像识别有了理论基础。

一个图像识别系统可分为三个主要部分：(1)图像信息的获取；(2)信息的加工和处理，抽取特征；(3)判断或分类。其框图如图(10-1)所示。

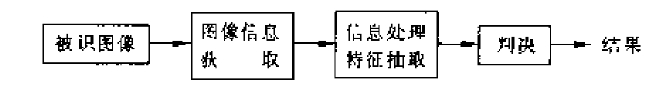


图 10-1 图像识别系统框图

第一部分相当于对被研究对象的调查和了解，从中得到数据和材料；对图像识别来说，就是把图片，底片，文字图形等用光电扫描设备变换为电信号，而对语音来说就可用话筒等设备

把声音变成电信号以备后序处理。第二部分相当于人们把调查了解到的数据材料进行加工、整理、分析、归纳，以去伪存真，去粗取精，抽出能反映事物本质的东西。当然，抽取什么特征，保留多少特征与采用何种判决有很大关系。第三部分是判决和分类，这相当与人们从感性认识上升到理性认识而做出结论的过程。第三部分与抽取特征的方式密切相关。它的复杂程度依赖于特征的抽取方式，例如：类似度、相关性、最小距离等。

模式识别的主要方法可分为两大类，这就是统计学方法和语言学方法。前一种方法是建立在被研究对象的统计知识上，也就是对图像进行大量的统计分析，抽出图像中本质的特征而进行识别。这是一种数学方法，它是受数学中的决策理论的启发而产生的识别方法。在这种方法中很大的力量用在抽取图像特征方面，也就是把图像大量的原始信息减缩为少数特征，然后再提取这些特征，把它作为识别的依据。另外一种是语言学法或句法结构识别法。它是立足于分析图像结构。把一个图像看成语言构造，例如一个英文句子，它是词和短语组成的并按一定的语法表达出来，其中最基本元素是单词。与此类似，图像是由一些直线、斜线、点、弯曲线及环等组成。剖析这些基本原素，看它们是以什么规则构成图像，这就是结构分析的课题。这些基本元素相当于句子中的单词，那些直线，曲线的组合相当于短语，它们全体如何构成图像就相当于语法规则。此时，图像识别就相当于检查图像所代表的某一类句型是否符合事先规定的语法，如果语法正确就识别出结果。由此可见，这种方法主要是利用了图像结构上的关系，这和统计学方法不同。

从上述两类方法看来，第一种方法没有利用图像本身的结构关系，第二类方法没有考虑图像在环境中受噪声的干扰。如果两者结合起来考虑可能会有新的识别方法，目前这方面的研究还不多。除此之外，基于模糊数学的发展，目前正在发展一种模糊识别法。这种方法较多地考虑了人的逻辑思维方法，方法较为独特，这种方法的研究得到了人们的关注。

模糊识别的应用较广，大致可有如下几个方面：

(1) 字符识别 (Character recognition)；

(2) 医学诊断 (Medical diagnosis)；

(3) 遥感(Remote Sensing)；

(4) 人脸和指纹鉴别 (Identification of human faces and fingerprints)；

(5) 污染(Pollution)；

(6) 自动检查和自动化 (Automatic inspection and Automation)；

(7) 可靠性(Reliability)；

(8) 社会经济 (Socio-economics)；

(9) 语音识别和理解(Speech understanding and recognition)；

(10)考古 (Archaeology)。

目前世界上已有一些较为完善的图像识别系统。这些系统无论从识别分析的功能来讲还是从处理速度上来说都较初期有很大的发展。例如，美国的 OLPARS (联机图像分析识别系统)能识别数字、字母及分析识别航空照片。英国的新产品 QUANTIMET 720 高速多功能图像分析系统可以观察由光学和显微镜获得的图像、照片、底片、电影、幻灯片及 X 光照片。能对图像进行各种测量及单独实时测量特征，数据由微计算机处理。日本的 OCR-ASPET/71 型识别系统能识别多种字体，每秒钟可识别 2000 字。英国的 IBM 1287 光学文字阅读机能识别 10 个阿拉伯数字，在邮局推广应用，误识率为 0.4%，拒识率为 1.1%。日本 NEC 公司研制的邮区编码信函分拣机能识别印刷体数字、字母，速度达 30000 件/小时。在医学中也有较多应用，

如一种5类白血球分类器可做到95%的正确分类，每分钟100个细胞。另外还有染色体自动分类，医学管理等方面也多有应用。

随着计算机技术的发展，模式识别的理论和方法得到进一步发展特别是图像识别这个领域近年来兴旺活跃，发展蓬勃。在某种意义上来说，图像识别已发展成为人与机器，自然科学和社会科学基础理论与技术应用之间的接口领域。目前，不仅研究单一功能的识别系统，而且在研制多功能的综合识别系统。如北方交通大学信息科学研究所会同清华大学、上海交通大学研究的"超级智能视听信息处理系统"就是一种多信息融合的处理系统，它的目的是利用多信息的融合技术，在模式识别中互相补充、互相借鉴，从而克服过去单一识别所面临的难以克服的困难，试图在模式识别领域有较大的突破。同时，该系统在当今颇为热门的人-机通过自然手段进行交互的领域也进行了有益的尝试。近年来国际上在这一领域给予了极大的重视，微软、Intel、IBM等大公司纷纷提出研究计划，所谓的"Multimodel"研究已形成了新的研究热点。与此同时，对有关图像识别的图像处理软件及新算法也受到极大的重视，如人工神经元网络、遗传算法等在模式识别研究中已取得了可喜的结果。现在，研制高性能、多用途的图像分析识别系统仍是有待我们努力解决的课题，随着生产与科学技术的发展，各个领域将会给模式识别技术以极大的注意，这一技术必将在我国的现代化建设中发挥作用。

10.2 统计模式识别法

统计模式识别的过程如图10-2所示。这是计算机识别的基本过程。数字化的任务是把图像信号变成计算机能够接受的数字信号。预处理的目的是去除干扰、噪声及差异，将原始信号变成适合于进行特征抽取的形式，然后，对经过预处理的信号进行特征抽取。最后进行判决分类，得到识别结果。为了进行分类，必须有图像样本。对样本图像进行特征选择及学习是识别处理中所必要的分析工作。

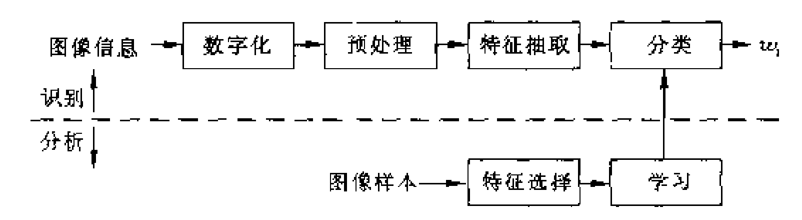


图10-2 统计模式识别法框图

10.2.1 决策理论方法

正如图10-2的框图所示，统计模式识别方法最终归结为分类问题。假如已抽取出N个特征，而图像可分为m类，那么就可以对N进行分类，从而决定未知图像属于m类中的哪一类。一般把识别模式看成是对N维空间中的向量$\boldsymbol{X}$进行分类，即：

$$\boldsymbol{X}=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \tag{10-1}$$

2

模式类别为$w_1,w_2,w_3,\cdots,w_m$。识别就是要判断$\boldsymbol{X}$是否属于w_i以及x_j属于w_i中的哪一类。在这个过程中主要解决两个问题：一是如何抽取特征，要求特征数N尽可能小而且对分类判断有效；二是假设已有了代表模式的向量，如何决定它属于哪一类，这就需要判别函数。例如，模式有$w_1,w_2,w_3,\cdots,w_m$共m个类别，则应有，$D_1(\boldsymbol{X}),D_2(\boldsymbol{X}),D_3(\boldsymbol{X}),\cdots,D_m(\boldsymbol{X})$，共$m$个判别函数。如果$\boldsymbol{X}$属于第$i$类，则有：

$$D_i(\boldsymbol{X}) > D_j(\boldsymbol{X}) \qquad j = 1,2,\cdots,m \quad j \neq i \tag{10-2}$$

在两类的分界线上，则有

$$D_i(\boldsymbol{X}) = D_j(\boldsymbol{X}) \tag{10-3}$$

这时$\boldsymbol{X}$既属于第i类，也属于第j类，因此这种判别失效。为了进行识别就必须重新考虑其他特征，再进行判别。问题的关键是找到合适的判别函数。

1．线性判别函数

线性判别函数是应用较广的一种判别函数。所谓线性判别函数是指判别函数是图像所有特征量的线性组合，即

$$D_i(\boldsymbol{X}) = \sum_{k=1}^{N} w_{ik}x_k + w_{i0} \tag{10-4}$$

式中w_i代表第i个判别函数；w_{ik}是系数或权；w_{i0}为常数项或称为阈值。在两类之间的判决界处有式(10-5)的形式：

$$D_i(\boldsymbol{X}) - D_j(\boldsymbol{X}) = 0 \tag{10-5}$$

该方程在二维空间中是直线，在三维空间中是平面，在N维空间中则是超平面。

$D_i(\boldsymbol{X})-D_j(\boldsymbol{X})$可以写成下式形式：

$$D_i(\boldsymbol{X}) - D_j(\boldsymbol{X}) = \sum_{k=1}^{N}(w_{ik} - w_{jk})x_k + (w_{i0} - w_{j0}) \tag{10-6}$$

其判决过程可如下进行：如果$D_i(\boldsymbol{X})>D_j(\boldsymbol{X})$或$D_i(\boldsymbol{X})-D_j(\boldsymbol{X})>0$，则$x\in w_i$；如果$D_i(\boldsymbol{X})<D_j(\boldsymbol{X})$或$D_i(\boldsymbol{X})-D_j(\boldsymbol{X})<0$，则$x\in w_j$。

用线性判别函数进行分类的是线性分类器。任何m类问题都可以分解为$(m-1)$个2类识别问题。方法是先把模式空间分为1类和其他类，如此进行下去即可。因此，最简单和最基本的是两类线性分类器。

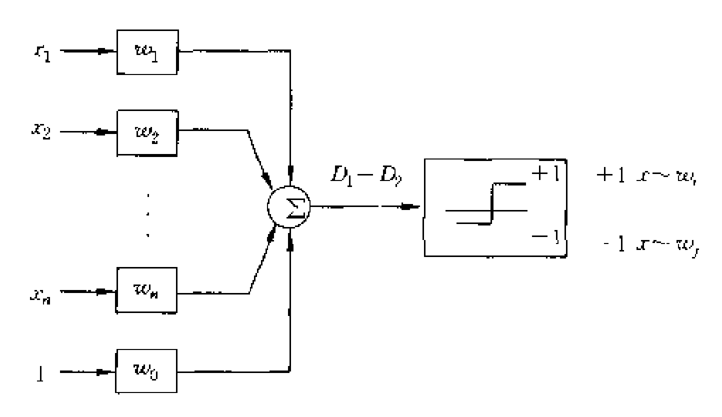


图10-3　两类线性分类器

分离两类的判决界由$D_1-D_2=0$表示。对于任何特定的输入模式必须判定D_1大还是D_2大。若考虑某个函数$D=D_1-D_2$，对于1类模式D为正，对于2类模式D为负。于是，只要处理与D相应的一组权的输入模式并判断输出符号即可进行分类。执行这种运算的分类器的原

理框图如图 10-3 所示。

在线性分类器中要找到合适的系数，以便使分类尽可能不出差错，惟一的办法就是试验法。例如，先设所有的系数为 1，送进每一个模式，如果分类有错就调整系数，这个过程就叫做线性分类器的训练或学习。例如，我们把 N 个特征 $\boldsymbol{X}$ 和 1 放在一起叫做 $\boldsymbol{Y}$，$N+1$ 个系数为 $\boldsymbol{W}$，即：

$$\boldsymbol{Y}=\begin{bmatrix}x_1\\x_2\\x_3\\\vdots\\x_N\\1\end{bmatrix}\qquad \boldsymbol{W}=\begin{bmatrix}w_1\\w_2\\w_3\\\vdots\\w_N\\w_{N+1}\end{bmatrix} \tag{10-7}$$

考虑分别属于两个不同模式类，$m=2$，此时，有两个训练集 T_1 和 T_2。两个训练集合是线性可分的，这意味着存在一个加权向量 $\boldsymbol{W}$，使得

$$\begin{cases}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}>0 & \boldsymbol{Y}\in T_1\\ \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}<0 & \boldsymbol{Y}\in T_2\end{cases} \tag{10-8}$$

式中 $\boldsymbol{Y}^{\mathrm{T}}$ 是 $\boldsymbol{Y}$ 的转置。

如果分类器的输出不能满足式(10-8)的条件，可以通过“误差校正”的训练步骤对系数加以调整。例如，如果第一类模式 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}$ 不大于零，则说明系数不够大，可用加大系数的方法进行误差修正。具体修正方法如下：

对于任一个 $\boldsymbol{Y}\in T_1$，若 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}\leqslant 0$，则

$$\boldsymbol{W}'=\boldsymbol{W}+\alpha\boldsymbol{Y} \tag{10-9}$$

对于任一个 $\boldsymbol{Y}\in T_2$，若 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}>0$，则

$$\boldsymbol{W}'=\boldsymbol{W}-\alpha\boldsymbol{Y} \tag{10-10}$$

通常使用的误差修正方法有固定增量规则，绝对修正规则及部分修正规则。固定增量规则是选择 α 为一个固定的非负数。绝对修正规则是取 α 为一个最小整数，它可使 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}$ 的值刚好大于零，即

$$\alpha=\text{大于}\frac{|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}|}{\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}}\text{的最小整数} \tag{10-11}$$

部分修正规则可取 α 为下式所决定的值

$$\alpha=\lambda\frac{|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{W}|}{\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}}\qquad 0<\lambda\leqslant 2 \tag{10-12}$$

2. 最小距离分类器

线性分类器中重要的一类是用输入模式与特征空间中作为模板的点之间的距离作为分类的准则。假设有 m 类，给出 m 个参考向量 $\boldsymbol{R}_1,\boldsymbol{R}_2,\cdots,\boldsymbol{R}_i$，$\boldsymbol{R}_i$ 与模式类 w_i 相联系。对于 $\boldsymbol{R}_i$ 的最小距离分类就是把输入的新模式 $\boldsymbol{X}$ 分为 w_i 类，其分类准则就是 $\boldsymbol{X}$ 与参考模式原型 $\boldsymbol{R}_1,\boldsymbol{R}_2,\cdots,\boldsymbol{R}_m$ 之间的距离，与哪一个最近就属于哪一类。$\boldsymbol{X}$ 与 $\boldsymbol{R}_i$ 之间的距离可表示为

$$|\boldsymbol{X}-\boldsymbol{R}_i|=\sqrt{(\boldsymbol{X}-\boldsymbol{R}_i)^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{R}_i)} \tag{10-13}$$

其中 $(\boldsymbol{X}-\boldsymbol{R}_i)^{\mathrm{T}}$ 是 $(\boldsymbol{X}-\boldsymbol{R}_i)$ 的转置。由式(10-13)得：

$$|\boldsymbol{X}-\boldsymbol{R}_i|^2=(\boldsymbol{X}-\boldsymbol{R}_i)^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{R}_i)$$

$$= X^{T}X - X^{T}R_i - R_i^{T}X + R_i^{T}R_i$$
$$= X^{T}X - (X^{T}R_i + R_i^{T}X - R_i^{T}R_i)$$

由此可设定最小距离判别函数 $D_i(X)$ 为

$$D_i(X) = X^{T}R_i + R_i^{T}X - R_i^{T}R_i \qquad i = 1,2,\cdots,m \tag{10-14}$$

由上边的判别函数可知，在分类中，如果 $X \in w_i$，则 $d(X,R_i) = \min$。由式(10-14)可见 $D_i(X)$ 是一个线性函数，因此，最小距离分类器也是一个线性分类器。在最小距离分类中，在决策边界上的点与相邻两类都是等距离的，这种方法就难于解决，此时必须寻找新的特征，重新分类。

这种分类还可以用决策区域来表示。例如，有二类问题 w_1, w_2，其模板分别为 R_1, R_2，当距离 $d(X,R_1) < d(X,R_2)$ 或者

$$\left[\sum_{i=1}^{n}(X_i - R_1)^2\right]^{\frac{1}{2}} < \left[\sum_{i=1}^{n}(X_i - R_2)^2\right]$$

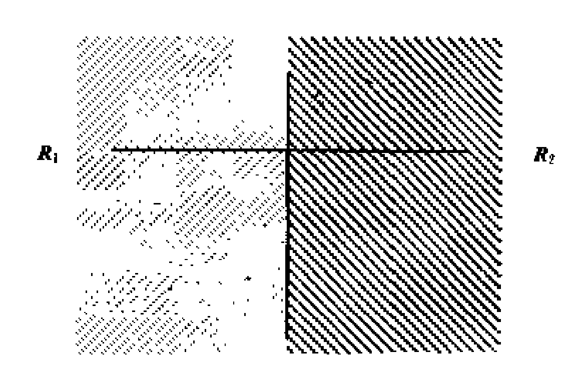


图 10-4　二类问题决策区域

则 $X \in w_1$，并可用决策区域表示，如图 10-4 所示。将模板 R_1, R_2 作连线，再作平分线，平分线左边为 R_1 区域，平分线右边为 R_2 区域，R_1、R_2 为决策区域，中间为决策面。在这种分类中，两类情况界面为线，决策区为两平面。对于三类情况，界面为超平面，决策区为半空间。

3. 最近邻域分类法

最近邻域分类法是图像识别中应用较多的一种方法。在最小距离分类法中，取一个最标准的向量作为代表。将这类问题稍微扩张一下，一类不能只取一个代表，把最小距离的概念从一个点和一个点间的距离扩充到一个点和一组点之间的距离。这就是最近邻域分类法的基本思路。设 $R_1, R_2, \cdots, R_m$ 分别是与类 $w_1, w_2, \cdots, w_m$ 相对应的参考向量的 m 个集合，在 R_i 中的向量为 R_i^k，即 $R_i^k \in R_i$，$k = 1,2,\cdots,l$，也就是

$$R_i = \{R_i^1 R_i^2 \cdots R_i^l\} \tag{10-15}$$

输入特征向量 X 与 R_i 之间的距离用下式表示：

$$d(X,R_i) = \min_{k=1,2,\cdots,l} |X - R_i^k| \tag{10-16}$$

这就是说，X 和 R_i 之间的距离是 X 和 R_i 中每一个向量的距离中的最小者。如果 X 与 R_i^k 之间的距离由式(10-14)确定，则其判别函数为

$$D_i(X) = \min_{k=1,2,\cdots,l}\{X^{T}R_i^k + (R_i^k)^{T}X - (R_i^k)^{T}R_i^k\} \tag{10-17}$$
$$i = 1,2,\cdots,m$$

设 $$D_i^k(\boldsymbol{X})=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{R}_i^k+(\boldsymbol{R}_i^k)^{\mathrm{T}}\boldsymbol{X}-(\boldsymbol{R}_i^k)^{\mathrm{T}}\boldsymbol{R}_i^k \tag{10-18}$$

则 $$D_i(\boldsymbol{X})=\min_{k=1,2,\cdots,l}\{D_i^k(\boldsymbol{X})\} \tag{10-19}$$

$$i=1,2,\cdots,m$$

其中 $D_i^k(\boldsymbol{X})$是特征的线性组合，决策边界将是分段线性的。例如，如图 10-5 所示，有一个两类判别问题，w_1 类的代表为 R_1^1 和 R_1^2，w_2 类的代表为 R_2^1,R_2^2,R_2^3。如果有一个模式送人识别系统，首先要计算它与每个点的距离，然后找最短距离。这种方法的概念简单，分段线性边界可以代表很复杂的曲线，也可能本来是非线性边界，现在可用分段线性来近似代替。

4. 非线性判别函数

线性判别函数很简单，但也有缺点。它对于较复杂的分类往往不能胜任。在较复杂的分类问题中就要提高判别函数的次数，因此根据问题的复杂性，可将判别函数从线性推广到非线性。非线性判别函数可写成下式形式：

$$\begin{aligned}D(x)=&\ w_0+w_1x_1+w_2x_2+\cdots+w_Nx_N\\&+w_{12}x_1x\ +w_{13}x_1x_3+\cdots+w_{1N}x_1x_N\\&+w_{11}x_1^2-w_{22}x_2^2+\cdots+w_{NN}x_N^2\\=&\ w_0+\sum_{k=1}^{N}w_{kk}x_k^2+\sum_{k=1}^{N}w_kx_k+\sum_{k=1}^{N}\sum_{l=1}^{N}w_{kl}x_kx_l\end{aligned} \tag{10-20}$$

式(10-20)是一个二次型判别函数。通常二次型判别函数的决策边界是一个超二次曲面。

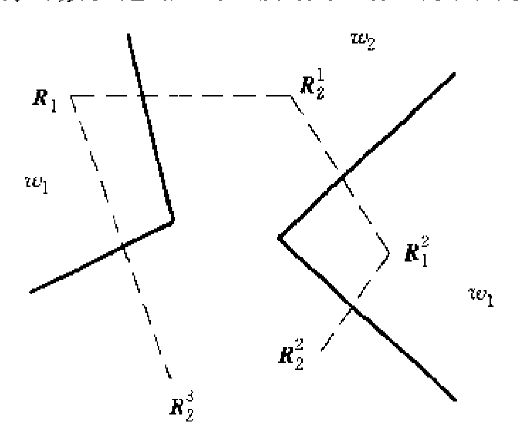


图 10-5　二类最近邻域分类

10.2.2　统计分类法

前边谈到的分类方法是在没有噪声干扰的情况下进行的，此时测得的特征确能代表模式。如果在抽取特征时有噪声，那么可能抽取的特征代表不了模式，这时就要用统计分类法。用统计方法对图像进行特征抽取、学习和分类是研究图像识别的主要方法之一，而统计方法的最基本内容之一是贝叶斯(Bayes)分析，其中包括贝叶斯决策方法、贝叶斯分类器、贝叶斯估计理论、贝叶斯学习、贝叶斯距离等等。

1. 贝叶斯公式

在古典概率中就已有为大家所熟悉的贝叶斯定理：

$$P(B_i|A)=\frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{n}P(B_j)P(A|B_j)} \tag{10-21}$$

式中 $B_1, B_2, \cdots, B_n$ 是 n 个互不相容的事件，$P(B_i)$是事件 B_i 的先验概率，$P(A|B_i)$是 A 在 B_i 已发生条件下的条件概率。贝叶斯定理说明在给定了随机事件 $B_1, B_2, \cdots, B_n$ 的各先验概率 $P(B_i)$及条件概率 $P(A|B_i)$时，可计算出事件 A 出现时事件 B_i 出现的后验概率$R(B_i|A)$。

贝叶斯公式常用于分类问题和参数估值问题中。假如设 X 表示事物的状态或特征的随机变量，它可以代表图像的灰度或形状等；设 w_i 代表事物类别的离散随机变量。对事物（比如是图像的亮度或形状）进行分类就可以用如下公式：

$$P(w_i|X) = \frac{P(X|w_i)P(w_i)}{\sum_i P(X|w_i)P(w_i)} \tag{10-22}$$

式中 $P(w_i)$称为 w_i 的先验概率，它表示事件属于 w_i 的预先粗略了解；$P(X|w_i)$表示事件属于 w_i 类而具有 X 状态的条件概率；$P(w_i|X)$叫做 X 条件下 w_i 的后验概率，它表示对事件 X 的状态作观察后判断属于 w_i 类的可能性。由式(10-22)可见，只要类别的先验概率及 X 的条件概率已知，就可以得到类别的后验概率。再加上最小误差概率或最小风险法则，就可以进行统计判决分类。

在参数估值问题中，贝叶斯公式中的二个变量常常为连续随机变量，如果写作变量 X 及参数 θ，则有如下之公式形式：

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)\mathrm{d}\theta} \tag{10-23}$$

通过式(10-23)，由参数的先验分布 $p(\theta)$及预先设定的条件分布 $p(X|\theta)$，即可求得参数的后验分布 $p(\theta|X)$。贝叶斯公式是参数估值的有力工具。

2. 贝叶斯分类法

假设有两类，每类用两种统计参数代表，即

$$\begin{aligned} w_1: &\quad P(w_1), p(X|w_1) \\ w_2: &\quad P(w_2), p(X|w_2) \end{aligned} \tag{10-24}$$

其中 $P(w_1), P(w_2)$是先验概率，$p(X|w_1), p(X|w_2)$是条件概率密度函数。在噪声的不确定性的影响下，每个模式已不能用一个向量来表示，因此，只能得到某一类模式的概率分布。

如果用贝叶斯规则的话，结果是

$$\begin{aligned} &\text{如果} \quad P(w_1)p(X|w_1) > P(w_2)p(X|w_2) \\ &\text{则有} \quad X \in w_1 \\ &\text{如果} \quad P(w_1)p(X|w_1) < P(w_1)p(X|w_1) \\ &\text{则有} \quad X \in w_2 \end{aligned} \tag{10-25}$$

显然，$P(w_i)p(X|w_i)$在这里起到了判别函数的作用。

在应用中，为方便起见，常取 $P(w_i)p(X|w_i)$的对数形式，即

$$\lg P(w_1)p(X|w_1) \leqslant \lg P(w_2)p(X|w_2) \tag{10-26}$$

也就是

$$\begin{cases} \lg \dfrac{p(X|w_1)}{p(X|w_2)} > \lg \dfrac{P(w_2)}{P(w_1)} & X \in w_1 \\ \lg \dfrac{p(X|w_1)}{p(X|w_2)} < \lg \dfrac{P(w_2)}{P(w_1)} & X \in w_2 \end{cases} \tag{10-27}$$

在两类问题中，分界面为

$$\lg P(w_1)p(X|w_1) - \lg P(w_2)p(X|w_2) = 0 \tag{10-28}$$

或者

$$\lg \frac{P(w_1)p(X|w_1)}{p(w_2)P(w_2)} = 0 \tag{10-29}$$

假如一个模式遵循正态分布，它的均值为 $\boldsymbol{M}_i$，协方差矩阵是 $\boldsymbol{K}_i$，设 $m=2$，可得到其决策分界面如下：

因为 $p(\boldsymbol{X}|w_i)$是正态分布，所以，

$$p(\boldsymbol{X}|w_1) = (2\pi)^{-\frac{N}{2}}|K_i|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{M}_i)^{\mathrm{T}}\cdot\boldsymbol{K}^{-1}(\boldsymbol{X}-\boldsymbol{M}_i)\right] \tag{10-30}$$

当 $i=1,2$ 时，按贝叶斯规则

如果

$$\lg\frac{p(\boldsymbol{X}|w_1)}{p(\boldsymbol{X}|w_2)} > \lg\frac{P(w_2)}{P(w_1)}，则 \boldsymbol{X}\in w_1 \tag{10-31}$$

由式(10-30)和式(10-31)可得到

$$-\frac{1}{2}\lg\frac{|\boldsymbol{K}_1|}{|\boldsymbol{K}_2|} - \frac{1}{2}[(\boldsymbol{X}-\boldsymbol{M}_1)^{\mathrm{T}}\boldsymbol{K}_1^{-1}(\boldsymbol{X}-\boldsymbol{M}_1)]$$

$$+\frac{1}{2}[(\boldsymbol{X}-\boldsymbol{M}_2)^{\mathrm{T}}\boldsymbol{K}_2^{-1}(\boldsymbol{X}-\boldsymbol{M}_2)] > \lg\frac{P(w_2)}{P(w_1)} \tag{10-32}$$

这时，两类间的决策边界是二次的。

如果两个协方差矩阵相同，即 $\boldsymbol{K}_1=\boldsymbol{K}_2=\boldsymbol{K}$，则

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{K}^{-1}(\boldsymbol{M}_1-\boldsymbol{M}_2) + \frac{1}{2}(\boldsymbol{M}_1+\boldsymbol{M}_2)^{\mathrm{T}}\boldsymbol{K}^{-1}(\boldsymbol{M}_1-\boldsymbol{M}_2) > \lg\frac{P(w_2)}{P(w_1)}$$

则有

$$\boldsymbol{X}\in w_1$$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{K}^{-1}(\boldsymbol{M}_1-\boldsymbol{M}_2) + \frac{1}{2}(\boldsymbol{M}_1+\boldsymbol{M}_2)^{\mathrm{T}}\boldsymbol{K}^{-1}(\boldsymbol{M}_1-\boldsymbol{M}_2) < \lg\frac{P(w_2)}{P(w_1)} \tag{10-33}$$

则有

$$\boldsymbol{X}\in w_2$$

在这种情况下，决策边界成为线性的。所以，求两类分类问题时，如果每类都是正态分布，但有不同的协方差矩阵，分界是二次函数，如果 N 很大，求 $\boldsymbol{K}^{-1}$相当麻烦。

除了上述方法外，也可以用最小风险来求其类别。

考虑 $x_1,x_2\cdots,x_N$ 是随机变量，对于每一类模式 w_i，$i=1,2,\cdots,m$，其 $p(\boldsymbol{X}|w_i)$及 w_i 出现的概率 $P(w_i)$都是已知的。以 $p(\boldsymbol{X}|w_i)$及 $P(w_i)$为基础，一个分类器成功的条件是要在误识概率最小的条件下来完成分类任务。我们可定义一个决策函数 $d(x)$，其中 $d(x)=d_i$ 表示假设 $X\in w_i$被接受。如果输入模式实际是来自 w_i，而作出的决策是 d_j，则可用 $L(w_i,d_j)$表示分类器引起的损失。条件风险为

$$r(w_i,d) = \int L(w_i,d)p(\boldsymbol{X}|w_i)\mathrm{d}\boldsymbol{X} \tag{10-34}$$

对于给定的先验概率集 $P=\{P(w_1),P(w_2),\cdots,P(w_m)\}$，平均风险为

$$R(P,d) = \sum_{i=1}^{m}P(w_i)r(w_i,d) \tag{10-35}$$

把式(10-34)代入式(10-35)并且令

$$r_x(P,d) = \frac{\sum_{i=1}^{m}L(w_i,d)P(w_i)p(\boldsymbol{X}|w_i)}{P(\boldsymbol{X})} \tag{10-36}$$

则

$$R(P,d)=\int P(X)r_x(P,d)\mathrm{d}\boldsymbol{X} \tag{10-37}$$

$r_x(P,d)$定义为对于给定的特征向量 $\boldsymbol{X}$,决策为 d 的后验条件平均风险。

问题在于选择适当的决策 d_i, $i=1,2,\cdots,m$,以使平均风险 $R(P,d)$取极小,或者使条件平均风险 $r(w_i,d)$的极大值取极小。这种使平均风险取极小的最优决策规则称为贝叶斯规则。

如果 d^* 是在使平均损失极小的意义上的最优决策,则

$$r_x(P,d^*)\leqslant r_x p(P,d) \tag{10-38}$$

即

$$\sum_{i=1}^{m}L(w_i,d^*)P(w_i)p(\boldsymbol{X}|w_i)$$

$$\leqslant\sum_{i=1}^{m}L(w_i,d)P(w_i)p(\boldsymbol{X}|w_i) \tag{10-39}$$

对于(0,1),损失函数为

$$L(w_i,d_j)=\begin{cases}0 & i=j\\ 1 & i\neq j\end{cases} \tag{10-40}$$

平均风险实际上也就是误识的概率。在这种情形下,贝叶斯规则是:

$$d^*=d_i,\ 则\ \boldsymbol{X}\in w_i$$

即

$$P(w_i)p(\boldsymbol{X}|w_i)\geqslant P(w_j)p(\boldsymbol{X}|w_j)$$

$$j=1,2,\cdots,m \tag{10-41}$$

3. 贝叶斯分类器

多类贝叶斯分类器如图 10-6 所示。其中 $p(\boldsymbol{X}|w_i)$与 $P(w_i)$的乘积就是第 i 类判别函数 $D_i(\boldsymbol{X})$。如果 $D_i(\boldsymbol{X})>D_j(\boldsymbol{X})$,对于一切 $i\neq j$ 的情况下,则分类器就把给定的一个特性量归于 w_i 类。

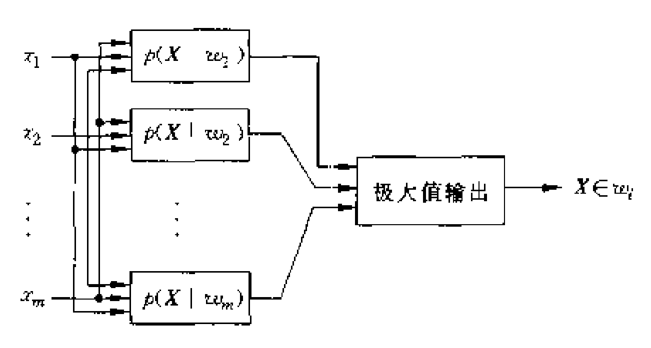


图 10-6 多类贝叶斯分类器

二类贝叶斯分类器如图 10-7 所示。在这类范畴的问题中,有时不制定二个判别函数 $D_1(\boldsymbol{X})$和 $D_2(\boldsymbol{X})$,而是定义一个判别函数:

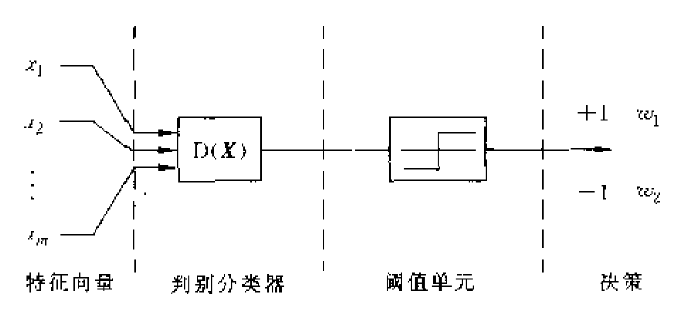


图 10-7 二类贝叶斯分类器

$$D(X) = D_1(X) - D_2(X) \tag{10-42}$$

若 $D(X)>0$,则决策 w_1,否则决策 w_2。

10.2.3 特征的抽取与选择

在模式识别中,确定判据是重要的。但是问题的另一面,即如何抽取特征也是相当重要的。如果特征找不对,分类就不可能准确。这好比医生看病,如果只注意病人穿什么衣服,头发的长短,就不会正确诊断。当然,特征是很多的,如果把所有的特征不分主次全都罗列出来,N 会很大,这也会给正确判断带来麻烦。例如,如图 10-8 所示。有两类模式,用两个特征 x_1,x_2 来表达。在 x_1 上的投影为 ab、cd,在 x_2 上的投影为 ef、gh。那么,由图可见,ac 这一段肯定是属于 w_1 的,bd 肯定是属于 w_2 的,但是 cd 段就难以分出属于哪一类。一种设想是把坐标轴作一个旋转,变成 y_1,y_2,此时不再去测量 x_1,x_2,而是去测量 y_1,y_2,如图 10-9 所示。由图可见,这时检测 y_1 当然也分不清,可是检测 y_2 就可以分得很清。这说明当作一变换后,y_2 是一个很好的特征。

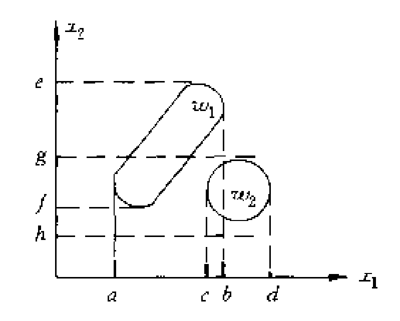


图 10-8 两类模式特征抽取之一

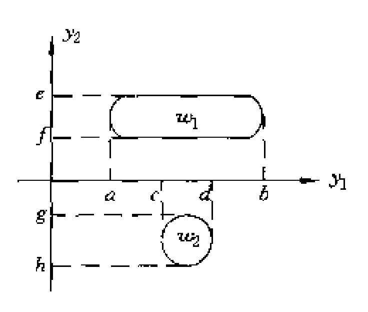


图 10-9 两类模式特征抽取之二

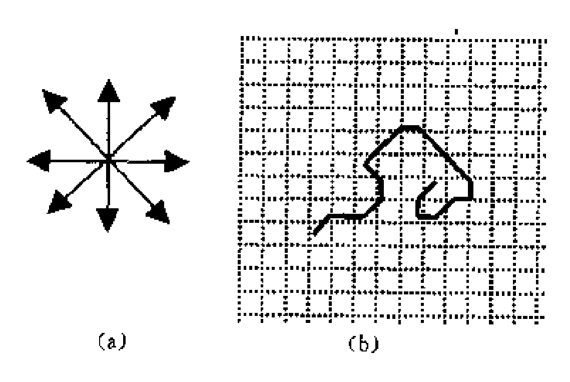
特征提取的方法是很多的。从一个模式中提取什么特征,将因不同的模式而异,并且与识别的目的、方法等有直接关系。常用的方法有离散直角坐标系中的弗里曼链码法。它可以方便地描述在离散直角坐标系中的曲线。图 10-10(a)是在 8 邻接定义下的弗里曼链码。位于坐标系内的任一条曲线便可用一个数字序列来表示。图 10-10(b)示出了一条曲线,若从 a 点出发可编出其链码如下:1 0 0 1 2 3 1 1 0 7 7 7 6 4 5 4 2 1。在提取边缘细条的过程中,会出现断线,因此,断线的接续是特征提取中的一个处理步骤。最基本的方法是利用膨胀和收缩技术。所谓膨胀是以二值图像内为 1 的像素为中心,强制性的把与其 4 邻接或 8 邻接的相邻像素都变成 1。如图 10-11 所示。

收缩方法是把值为 0 的像素作为中心,强制性地把与其 4 邻接或 8 邻接的相邻像素变成 0。这样连续膨胀 n 次,再连续收缩 n 次,就可以把断线长度为 $2n$ 以内的线接续起来。

接续断线的另一种方法是山脊线寻迹法。具体做法是使用某种方法找出直到点(x,y)为止的一段山脊线,接着判断点$(x+1,y+1)$,$(x+2,y+2)$等点是否也位于该山脊线的延长线上。判断的标准就是看这些点的微分值是否足够大,这些点周围的灰度变化斜率最大的方向是与线的延长线方向垂直。这些点与周围延长线方向成直角方向上的点相比灰度值是否为极大值等等。这种方法碰到折点及分枝点比较难于判断。

在特征提取中,关于线的检测及表达方法有最小二乘法曲线拟合法,霍夫变换法等等。在进行线提取时,往往不是简单地用一些直线段把检测出来的点连接起来就行了,而是希望用某个数学方程式所描述的曲线去逼近检测出来的点列。这种用数学方程式去近似图像中各种线

条的方法称为曲线拟合。



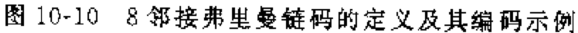


图 10-10 8 邻接弗里曼链码的定义及其编码示例

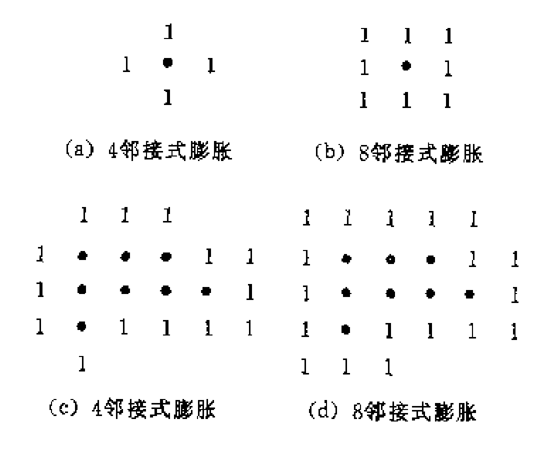


图 10-11 膨胀操作

最简单的曲线拟合是用直线方程去近似所给出的点列,这种方程有 $y=g(x)$ 之形式。在拟合处理中自然需要一定的标准去评价该方程与点列的近似程度,常用的评价标准是观察直线方程所代表的直线与点列之间的距离大小。对距离大小有不同的定义,常用的有:

$$d = \sum_{i=0}^{M} |y_i - g(x_i)| \tag{10-43}$$

$$d = \sum_{i=0}^{M} [y_i - g(x_i)]^2 \tag{10-44}$$

$$d = \max_i \{ |y_i - g(x_i)| \} \tag{10-45}$$

式(10-43)是以直线方程与各点之差的绝对值的和为最小作为评价标准;式(10-44)是以差的平方和最小作为评价标准,通常称为最小二乘法判决函数;式(10-45)是以差值中的最大值是否小于某一标准进行评价。

霍夫变换也广泛用于线检测,它的概念已在第 8 章中作了介绍,在此不再赘述。除此之外,用于线检测的特征提取方法还有很多,如用曲率作为曲线的特征,曲线分割,距离变换、骨骼化及细化等均是在特征提取处理中常用的方法。

在图 10-8 和 10-9 所说明的特征提取的例子中,用坐标旋转的方法得到了既少又好的特征。空间坐标的旋转就是特征空间的线性变换。空间怎样变换才能找到较好的特征呢?其普遍的方法是把每一类的协方差矩阵变成对角形矩阵,在变换后的矩阵中取其特征向量及与其相对应的特征值,然后,把特征向量按其特征值的大小排列起来。特征值大的那个特征向量就是最好的特征。另外,在变换后的空间中,如果有 m 个彼此关联的特征,可采用前 n 个最大特征值对应的特征向量作为特征,这样既可保证均方误差最小,又可大大减少特征的数目。

另外一个途径是寻找一种变换,使同一类向量靠得更近些,以便把它聚合到一起。在这种思想指导下,可以找每一类点与点之间的距离,使它最小化。这样作是应用特征值最小的那些特征向量。

假定有两类模式,测量两种特征都是正态分布,均值是 m_1 和 m_2。这两个分布离得越远越容易识别。所谓离得远不一定是均值相关较远。在这种情况下,不能用点与点间的距离,也不是点与一组点间的距离,而是两个分布间的距离,这是一个统计距离。如果在统计意义上两类离得远就容易识别。如果有 M 个特征,就要计算它们的统计距离,哪个特征上的统计距离最远,哪个特征就最好。一般计算统计距离的方法有许多。例如:贝叶斯误差概率;疑义度或仙农

熵;贝叶斯距离;广义柯尔莫哥洛夫距离等等。

另外,在 m 很大时,同时分开 m 类比较复杂。这时不如采用树状分类结构,每次分两三类,逐次细分。当然,寻找一个最好的树也并非容易。

10.3 句法结构模式识别

统计决策识别法是模式识别中应用较广的一种方法。它的基本做法是首先从待识别模式中提取特征参数,然后用这些特征参数把模式表达为特征空间中的点,然后再根据各点之间的距离进行分类和识别。这种识别方法对结构复杂、形式多变的模式来说存在着一系列的困难。首先对比较复杂的模式需要较多的特征才能描述它,而特征提取是比较困难的环节,对于同一模式往往有不同的抽取方法,就目前来看尚没有统一的理论依据。其次,简单的分类并不能代表识别,对于复杂的模式,识别的目的并不是仅仅要求把它分配到某一类别中去,而且还要对不同的对象加以描述,在这方面统计决策法就有极大的局限性。近几年发展起来的句法结构模式识别法主要着眼于模式结构,采用形式语言理论来分析和描述模式结构,因此,它具有统计识别法所不具备的优点。所以句法结构模式识别法是一种颇受重视的近代识别方法。

10.3.1 形式语言概述

所谓句法结构就是将一个复杂的模式一部分一部分地加以描述,将复杂的模式分成若干子模式,如此分下去直至最简单的子模式(或称基元)为止。这是一种树状结构的表示方法。这种方法类似于语言分析中的句法结构,因此,把这种着眼于结构的模式识别方法叫做结构模式识别。在结构模式识别法中句法结构是借助于形式语言(数理语言)进行描述的。

1. 形式语言的几个基本定义

(1) 字母表(词汇表)

字母表在形式语言中被定义为与问题有关的符号集。例如:

$$V_1 = \{A,B,C,\cdots,Z\}$$

$$V_2 = \{a,b,c,d\}$$

$$V_3 = \{0,1\}$$

$$V_4 = \{I,go,to,you,like,in\}$$

等等都是字母表的例子。

(2) 句子(链)

这是由字母表中的符号组成的有限长的符号串。例如:

ABBA,0 1 0 1 1 0 1 1,I,like,horse 等均是句子。

(3) 语言

语言是由字母表组成的句子的集合。例如:

$$L_1 = \{a,aa,aba,aabba\}$$

$$L_2 = \{a^n b^m a^k \mid n,m,k = 0,1,2,\cdots\}$$

均是句子,它们的字母表为 $V=\{a,b\}$。语言可以分为有限的和无限的,上例中 L_1 是有限的,L_2 是无限的。

(4) 文法

文法是一种语言中构成句子所必须遵守的规则的有限集。

(5) V^*

它是由字母表的符号组成的所有句子的集合,当然,也包括空句子在内。

(6) $V^+=V^*-\lambda$

它表示不包括空句子在内的所有句子集合。例如:

$$V=\{0,1\}$$
$$V^*=\{\lambda,0,1,00,01,10,11,000,\cdots\}$$
$$V^{-}=\{0,1,00,01,10,11,000,\cdots\}$$

以上是形式语言中的基本定义。利用上述定义可以分析一个英文句子,如"The boy moves quickly",利用形式语言可把句子表示成图 10-12 所示的树形结构。

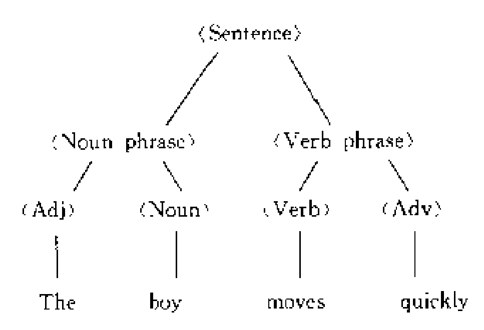


图 10-12　语言分析中的树状结构

2. 树状结构中语言分析的四个基本要素

(1) 终止符

例如:树状结构中的"The, boy, moves, quickly"均是终止符。

(2) 非终止符

例如:〈Sentence〉,〈Noun phrase〉,〈Verb phrase〉是非终止符。

(3) 产生式(或称为重写规则)

例如:〈Sentence〉→〈Noun phrase〉〈Verb phrase〉　(r_1)

〈Noun phrase〉→〈Adj〉〈Noun〉　(r_2)

〈Verb phrase〉→〈Verb〉〈Adv〉　(r_3)

〈Adj〉The　(r_4)

〈Noun〉→boy　(r_5)

〈Verb〉→moves　(r_6)

〈Adv〉→quickly　(r_7)

其中 $r_1, r_2, \cdots, r_7$ 是序号。

(4) 起始符

例如:〈Sentence〉

以上四元素就构成了文法,有了文法就可以构成句子,每一个句子都必须遵循文法规则。以上述句子为例其产生过程如下:

〈Sentence〉⇒〈Noun phrase〉〈Verb phrase〉

⇒〈Adj〉〈Noun〉〈Verb phrase〉

→The〈Noun〉〈Verb phrase〉

⇒The boy 〈Verb phrase〉

→The boy 〈Verb〉〈Adv〉

$\Rightarrow$The boy moves ⟨Adv⟩

$\Rightarrow$The boy moves quickly

其中⇒表示变换操作。

3. 短语结构文法

由前边的四元素可以构成短语结构文法。如第 8 章所述，短语结构文法为一个四元式。即

$$G = (V_N, V_T, P, S)$$

其中 V_N 为非终止符；V_T 为终止符；P 为产生式；S 为起始符，$S \in V_N$。

根据产生式形式的不同，又可分为不同的文法类型，这就是所谓的霍金斯分类。

(1) 0 型文法

产生式的一般形式为 $\alpha \rightarrow \beta$(α 在 V^+中，β 在 V^* 中)。在这种类型中，产生式没有限制，所以又称作无约束文法，这种文法无法处理。

(2) 1 型文法(上下文敏感文法)

产生式为 $\alpha_1 A \alpha_2 \rightarrow \alpha_1 B \alpha_2$，其中 $\alpha_1, \alpha_2, B \in V^*$，$A \in V_N$，$B \neq \lambda$，$B$ 代替 A 的条件必须是上下文为 α_1 和 α_1，即和上文 α_1 及下文 α_2 有关。

(3) 2 型文法(上下文无关文法)

产生式为 $A \rightarrow B$，其中 $A \in V_N$，$B \neq \lambda$，$B \neq A$。这种文法不考虑出现 A 的上下文就可以用 B 代替 A，因此称为上下文无关文法。

(4) 3 型文法(正规文法或有限状态文法)

产生式为 $A \rightarrow \alpha B$(或 $A \rightarrow B\alpha$)，$A \rightarrow \alpha$ 式中 $A, B \in V_N$，$\alpha \in V_T$。

以上四种文法的关系如图 10-13 所示。这种关系可概括如下：正规语言一定是上下文无关语言，上下文无关语言也一定是上下文敏感的语言。反之，能认识上下文无关的语言就能认识正规语言，能认识上下文敏感语言的机器也一定能认识上下文无关和正规语言。

除了上述文法外还有加以修改后的文法，如：程序文法、标号文法、转换网络文法等等。

对于各种不同的语言，如何去识别呢？在识别过程中，每一种语言都与某一特定的识别器相对应。与 0 型语言相对应的识别器是图灵机；与 1 型语言相对应的识别器是线性约束自动机；与 2 型语言相对应的识别器是推下自动机；正规型语言与有限自动机相对应。

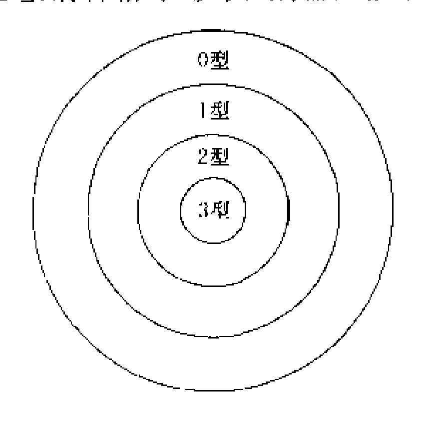


图 10-13 四种文法的关系

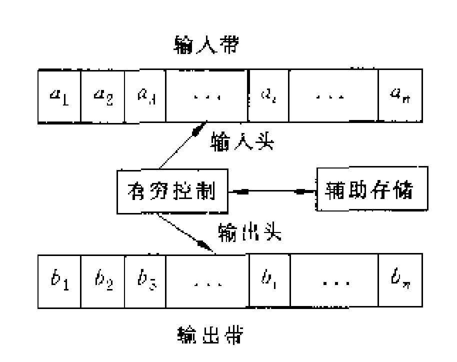


图 10-14 自动机模型

自动机(或叫抽象信息转换器)的模型如图 10-14 所示。它是由输入带，输出带，辅助存储器和一个具有有穷个规则的控制器组成的装置。通常把没有输入带的自动机叫做生成器，把没有输出带的自动机叫做识别器。输入带和输出带都假定是被分成一个个的方格，每个方格刚好印一个符号。输入头可以在带上左右移动，每次移动一个方格，也可以原地不动。有时也可以

擦去原来的符号并印上一个新的符号。输出头一次可以印一系列符号，但只允许向右移动。自动机的辅助存储器可以是任何类型的数据存储器。有穷控制对应着规则的集合，而规则描述了系统中的信息根据当前输入符号及存储器中当前存取的信息的变化规律。

4．有限状态自动机

有限状态自动机只能接受所有由有限状态文法（或称为正规文法、正则文法、右线性文法）所定义的语言。

一个确定性的有限状态自动机是一个五元式：

$$M=(\boldsymbol{\Sigma},\boldsymbol{Q},\delta,q_0,\boldsymbol{F})$$

其中：$\boldsymbol{\Sigma}$ 为输入符号的有限集合；$\boldsymbol{Q}$ 为状态的有限集合；q_0 为初始状态；$\boldsymbol{F}$ 为终止状态集合，$\boldsymbol{F}\subseteq\boldsymbol{Q}$；$\delta$ 为状态转移函数，是从 $\boldsymbol{Q}\times\boldsymbol{\Sigma}$ 到 $\boldsymbol{Q}$（下一状态函数）的映射。

一个非确定的有限状态自动机是一个五元式：

$$M=(\Sigma,\boldsymbol{Q},\delta,q_0,\boldsymbol{F})$$

其中：Σ 为输入符号有限集合；$\boldsymbol{Q}$ 为状态有限集合；q_0 为初始状态；$\boldsymbol{F}$ 为终止状态集；δ 为 $\boldsymbol{Q}\times\Sigma$ 到 $\boldsymbol{Q}$ 的子集的一个映射。非确定性有限状态自动机与确定性有限状态自动机的区别在于非确定性自动机可以从一个状态转移到若干个状态，即 $\delta(q,a)=\{q_1,q_2,\cdots,q_m\}$。

有限状态自动机如图 10-15 所示。有限控制处于 $\boldsymbol{Q}$ 中的一个状态，它以顺序方式从左到右地从输入带上读出符号。开始时，有限控制处在状态 q_0，并从最左面的符号扫描。$\delta(q,a)=q'$、q，$q'\in\boldsymbol{Q}$，$a\in\boldsymbol{\Sigma}$ 它们所表示的意思是：自动机 M 处于状态 q，扫描输入一个符号 a，转移到状态 q'，输入头向右移一格。这种映射常用状态转移图来表示。图 10-16 便是 $\delta(q,a)=q'$ 的状态转移图。

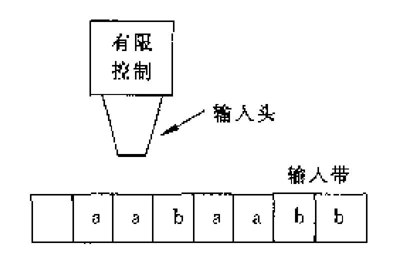


图 10-15　有限状态自动机

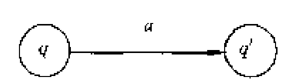


图 10-16　$\delta(q,a)=q'$ 的状态转移图

例 1　给定一个有限状态自动机，

$$M=(\boldsymbol{\Sigma},\boldsymbol{Q},\delta,q_0,\boldsymbol{F})$$

其中：$\boldsymbol{\Sigma}=\{0,1\}$；$\boldsymbol{Q}=\{q_0,q_1,q_2,q_3\}$；$\boldsymbol{F}=\{q_0\}$，$M$ 的状态转移图如图 10-17 所示。M 所接受的典型句子是 1 0 1 1 0 1，此时 $\delta(q_0,1\ 0\ 1\ 1\ 0\ 1)=q_0\in\boldsymbol{F}$。

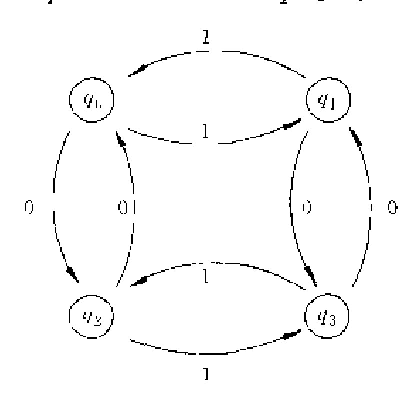


图 10-17　例 1 的自动机状态转移图

例 2　给定一个非确定的有限状态自动机，

$$\boldsymbol{M} = (\boldsymbol{\Sigma}, \boldsymbol{Q}, \boldsymbol{\delta}, q_0, \boldsymbol{F})$$

其中：

$$\boldsymbol{\Sigma} = \{0,1\}$$
$$\boldsymbol{Q} = \{q_0, q_1, q_2, q_3, q_4\}$$
$$\boldsymbol{F} = \{q_2, q_4\}$$

以及

$$\boldsymbol{\delta}(q_0,0) = \{q_0,q_3\}$$
$$\boldsymbol{\delta}(q_1,0) = \phi$$
$$\boldsymbol{\delta}(q_2,0) = \{q_2\}$$
$$\boldsymbol{\delta}(q_3,0) = \{q_4\}$$
$$\boldsymbol{\delta}(q_4,0) = \{q_4\}$$
$$\boldsymbol{\delta}(q_0,1) = \{q_0,q_1\}$$
$$\boldsymbol{\delta}(q_1,1) = \{q_2\}$$
$$\boldsymbol{\delta}(q_2,1) = \{q_2\}$$
$$\boldsymbol{\delta}(q_3,1) = \phi$$
$$\boldsymbol{\delta}(q_4,1) = \{q_4\}$$

其状态转移图如图 10-18 所示。由图可以看出被 $\boldsymbol{M}$ 接受的典型句子是 0 1 0 1 1，因为 $\boldsymbol{\delta}(q_0,01011)=q_2\in\boldsymbol{F}$。

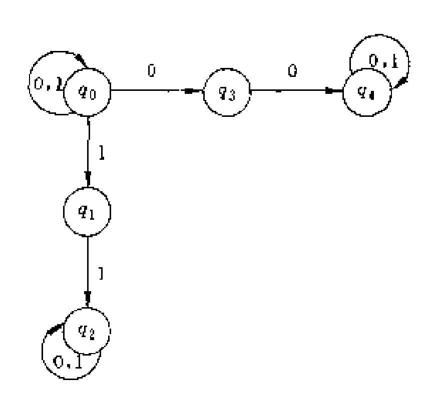


图 10-18　例 2 中自动机状态转移图

关于正规文法与有限状态自动机之间的关系可用下述定理来说明。

设 $G=(V_N,V_T,P,S)$是一正规文法，则存在一个有限状态自动机 $\boldsymbol{M}=(\boldsymbol{\Sigma},\boldsymbol{Q},\boldsymbol{\delta},q_0,\boldsymbol{F})$，且具有性质 $T(\boldsymbol{M})=L(G)$，其中：$\boldsymbol{\Sigma}=V_T$；$\boldsymbol{Q}=V_N\cup\{\boldsymbol{T}\}$；$q_0=S$；如果 P 包含产生式 $S\rightarrow\lambda$，则 $\boldsymbol{F}=\{S,T\}$，否则 $\boldsymbol{F}=\{T\}$；如果 $B\rightarrow a$ 在 P 中，$B\in V_N$，$a\in V_T$ 那么状态 T 在 $\boldsymbol{\delta}(B,a)$中；$\boldsymbol{\delta}(B,a)$包含所有 $C\in V_N$ 使 $B\rightarrow aC$ 在 P 中以及 $\boldsymbol{\delta}(T,a)=\phi$，对于每个 $a\in V_T$。

另外，给定一个有限状态自动机 $\boldsymbol{M}=(\boldsymbol{\Sigma},\boldsymbol{Q},\boldsymbol{\delta},q_0,\boldsymbol{F})$，则存在一个有限状态文法 $G=(V_N,V_T,P,S)$，且 $L(G)=T(A)$，其中：$V_N=\boldsymbol{Q}$；$V_T=\Sigma$；$S=q_0$；如果 $\boldsymbol{\delta}(B,a)=C$，$B,C\in\boldsymbol{Q}$，$a\in\boldsymbol{\Sigma}$，则 $B\rightarrow aC$ 在 P 中；如果 $\boldsymbol{\delta}(B,a)=C$，且 $C\in\boldsymbol{F}$，$B\rightarrow a$ 在 P 中。

例：给定正规文法于下：

$$G = (V_N, V_T, P, S)$$

$$
\begin{aligned}
&V_N = \{S, B\} \\
&V_T = \{a, b\} \\
&P: S \to aB \\
&\quad\; B \to aB \\
&\quad\; B \to bS \\
&\quad\; B \to a
\end{aligned}
$$

可构成一个非确定的有限状态自动机：

$$M = (\boldsymbol{\Sigma}, \boldsymbol{Q}, \boldsymbol{\delta}, q_0, \boldsymbol{F})$$

并且

$$T(\boldsymbol{M}) = L(G)$$

其中

$$
\begin{aligned}
&\boldsymbol{\Sigma} = V_T = \{a, b\} \\
&\boldsymbol{Q} = V_N \cup \{T\} = \{S, B, T\} \\
&q_0 \in S \\
&\boldsymbol{F} = \{T\}
\end{aligned}
$$

δ给出如下：

$\boldsymbol{\delta}(S, a) = \{B\}$，由于 $S \to aB$ 在 P 中

$\boldsymbol{\delta}(S, b) = \phi$

$\boldsymbol{\delta}(B, a) = \{B, T\}$，由于 $B \to aB, B \to a$ 在 P 中

$\boldsymbol{\delta}(B, b) = \{S\}$，由于 $B \to bS$ 在 P 中

$\boldsymbol{\delta}(T, a) = \delta\{T, b\} = \phi$

也可构成一个确定的有限状态自动机：

$\boldsymbol{M}' = (\boldsymbol{\Sigma}', \boldsymbol{Q}', \boldsymbol{\delta}', q_0', \boldsymbol{F}')$（等价于 $\boldsymbol{M}$）

$\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} = \{a, b\}$

$\boldsymbol{Q}' = \{\phi, [S], [B], [T], [S, B], [S, T], [B, T], [S, B, T]\}$

$q_0' = [S]$

$\boldsymbol{F}' = \{[T], [S, T], [B, T], [S, B, T]\}$

$\boldsymbol{\delta}'([B], a) = [B]$

$\boldsymbol{\delta}'([S], b) = \phi$

$\boldsymbol{\delta}'([B], a) = [B, T]$

$\boldsymbol{\delta}'([B], b) = [S]$

$\boldsymbol{\delta}'([B, T], a) = [B, T]$

$\boldsymbol{\delta}'([B, T], b) = [S]$

$\boldsymbol{\delta}' = (\phi, a) = \boldsymbol{\delta}'(\phi, b) = \phi$

还有 $\boldsymbol{\delta}'$ 的其他转移规则，但 $\boldsymbol{M}'$ 不会达到 ϕ，$[S]$，$[B]$ 及 $[B, T]$ 以外的状态。

5. 推下自动机

上下文无关的语言可以由推下自动机来接受。这种自动机与有限状态自动机不同之处在于多加了一个推下存储器，按照先入后出的规则存入或取出符号。一个推下自动机是一个七元式：

$$M_p = (\boldsymbol{\Sigma}, \boldsymbol{Q}, \boldsymbol{\Gamma}, \boldsymbol{\delta}, q_0, Z_0, \boldsymbol{F})$$

其中：Σ 为输入符号有限集；Q 为状态的有限集；Γ 为是下推符号有限集；q_0 为初始状态；Z_0 为最初出现在下推存储器中的起始符，$Z_0 \in \Gamma$；F 为终止状态集；δ 为从 $Q\times(\Sigma\cup\{\lambda\})\times\Gamma$ 到 $Q\times\Gamma^*$ 的有限子集的映射。

$$\delta(q,a,Z) = \{(q_1,r_1),(q_2,r_2),\cdots,(q_m,r_m)\}$$

$$q_1,q_2,\cdots,q_m \in Q \quad a \in \Sigma \quad Z \in \Gamma \quad r_1,r_2,\cdots,r_m \in \Gamma^*$$

上述公式可解释如下：推下自动机在状态 q，其输入符号为 a，下推存储的顶上的符号为 Z 时，对于任何 i，$1\leqslant i\leqslant m$，将进入状态 q_i，用 r_i 代替 Z，而且输入头前进一个符号。当 Z 由 r_i 代替时，r_i 最左边的符号将处于下推存储的最高处，而最右边的符号处于最低处。

推下自动机的原理如图 10-19 所示。

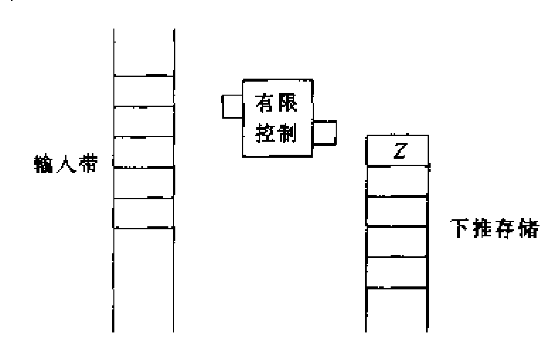


图 10-19　推下自动机原理模型

在推下自动机中，对于下列两种情况被认为 X 被接受：

$$T(M_p) = \{X|_{X\in\Sigma^*};\delta(q_0,X,Z_0) = (q_f,r),q_f \in F\}$$

即从初始状态开始，转移至终止状态集合，不管推下存储器的内容如何，此时则认为 X 被接受了。

$$N(M_p) = \{X|_{X\in\Sigma^*};\delta(q_0,X,Z_0) = (q_f,\lambda)\}$$

即不管 q_f 是否在 F 内，只要推下存储器变为空的，就认为 X 被接受。

上下文无关语言与推下自动机 M_p 之间的关系可以用以下定理表示：

1）如 L 是上下文无关语言，则存在一个推下自动机 M_p，使得 $L=N(M_p)$。

2）对某些推下自动机 M_p 而言，如果 L 是 $N(M_p)$，并且仅当对某些推下自动机 M_p' 而言，L 是 $T(M_p')$。

3）对某些推下自动机 M_p 而言，如果 L 是 $N(M_p)$，则 L 是上下文无关的语言。对于上下文无关文法 G，可以构造一个推下自动机 M_p，使得 $N(M_p)=L(G)$。

例　有推下自动机 M_p，

$$M_p = (\Sigma,Q,\Gamma,\delta,q_0,Z_0,F)$$

其中

$$\Sigma = \{0,1\}$$

$$Q = \{q_0,q_1,q_2\}$$

$$\Gamma = \{2,0\}$$

$$Z_0 = \{Z\}$$

$$F = \{q_0\}$$

$$\delta(q_0,0,Z) = (q_1,02)$$

$$\delta(q_1,0,0) = (q_1,00)$$
$$\delta(q_1,1,0) = (q_2,\lambda)$$
$$\delta(q_2,1,0) = (q_2,\lambda)$$
$$\delta(q_2,\lambda,Z) = (q_0,\lambda)$$

令 $X=0\ 0\ 1\ 1$,则有

$$(q_0,Z) \xrightarrow{0} (q_1,02) \xrightarrow{0} (q_1,002) \xrightarrow{1}$$
$$(q_2,02) \xrightarrow{1} (q_2,2) \xrightarrow{\lambda} (q_0,1)$$

需要指出的是,正规文法产生正规语言,一定能找到一个确定的有限自动机接受它。而上下文无关文法不一定能找到一个确定的推下自动机去接受同一语言。因此,对上下文无关的语言来说,非确定的推下自动机有多种选择,在程序上去实现要依次去试,这叫结构分析或剖析。

6. 图灵机和线性约束自动机

图灵机可接受 0 型语言,线性约束自动机可接受 1 型语言。图灵机的基本模型如图 10-20 所示。它由带有一个读写头的有限控制部分和输入带组成。带的右端是无限的。开始时,对于某一个有限的 n,最左边的 n 个方格为输入链所占据,其余无穷多个方格为空白。

一个图灵机是一个六元式:

$$\boldsymbol{T} = (\boldsymbol{\Sigma},\boldsymbol{Q},\Gamma,\delta,q_0,\boldsymbol{F})$$

其中:$\boldsymbol{Q}$ 为状态的有限集;Γ 为带上符号有限集,这些符号中包括一个空白符号 $\boldsymbol{B}$;$\boldsymbol{\Sigma}$ 为输入符号集,是 Γ 的不包括 $\boldsymbol{B}$ 的子集;q_0 为初始状态,$q_0\in\boldsymbol{Q}$;$\boldsymbol{F}$ 为终止状态集,$\boldsymbol{F}\subseteq\boldsymbol{Q}$;$\delta$ 为从 $\boldsymbol{Q}\times\Gamma$ 到 $\boldsymbol{Q}\times(\Gamma-\{\boldsymbol{B}\})\times\{L,R\}$ 的一个映射。

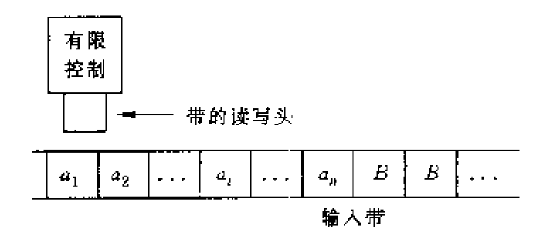


图 10-20 基本图灵机模型

图灵机与 0 型语言的关系有如下定理:

1)如果 L 是由 0 型文法产生的语言,则 L 会被一个图灵机所接受。

2)如果 L 被一个图灵机所接受,则 L 是由一个 0 型文法产生。

一个线性约束自动机是一个六元式:

$$\boldsymbol{M} = (\boldsymbol{\Sigma},\boldsymbol{Q},\Gamma,\delta,q_0,\boldsymbol{F})$$

其中:Σ 为是输入符号集,$\boldsymbol{\Sigma}\subseteq\Gamma$;$\boldsymbol{Q}$ 是状态有限集;q_0 为是初始状态,$q_0\in\boldsymbol{Q}$;$\boldsymbol{F}$ 为终止状态集;$\boldsymbol{F}\subseteq\boldsymbol{Q}$。$\Sigma$ 包含两个特殊符号 $\not{C}$ 和 $\$$,它们分别是输入链左边及右边的结束标志,其功能是防止读写头离开带子上输入出现的部分。

由定义可见,线性约束自动机实质上是永远不会离开带子上输入所在部分的一个非确定图灵机。上下文有关的语言与线性约束自动机的关系可由下列定理描述:

1) 如果 L 是上下文有关语言,则 L 由一个(非确定)线性约束自动机所接受;

2) 如果 L 由一个线性约束自动机所接受,则 L 是上下文有关的语言。

7. 结构分析(剖析)

结构分析的任务是给出一个由符号组成的句子 X 后,如何从起始符 S 推导出 X,也就是

如何构造一棵 X 的剖析树。当给定一个文法 G 和一个句子 X 时，要求回答"$X \in L(G)$?"的问题。根据文法产生的规则，如果能找到一棵产生该句子的树，那么 $X \in L(G)$，同时也知道了树状结构。

寻找剖析树的文法有多种，一种是从上往下的过程，找左边吻合的产生式换成右边的符号；另外一种文法是从下往上的过程，找右边的吻合的产生式换成左边的符号；此外，还有其他方法，如希克法及厄利剖析法等。

10.3.2 句法结构方法

句法方法的模式识别系统框图如图 10-21 所示。它由识别及分析两部分组成。识别部分包括预处理、基元抽取和结构分析。预处理主要包括编码、滤波、复原、增强及缝隙填补等一系列操作。基元抽取包括分割、特征(基元)抽取。这部分在分割的过程中抽取基元并显示基元相互关系，以便利用子模式进行描述。基元一定是模式的一部分，与统计识别中特征提取稍有不同，基元的选择要考虑容易识别，所以基元不一定是模式中最小的基本元素。基元的选择要尽可能少，而且容易被识别。

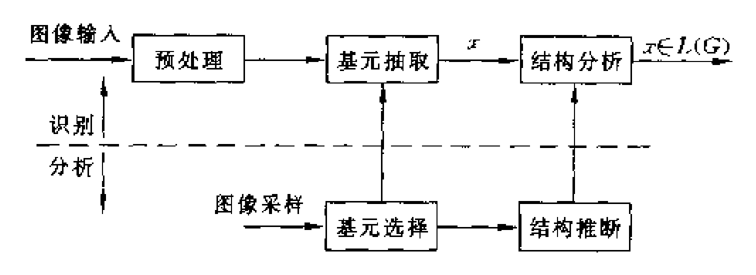


图 10-21 句法方法模式识别系统框图

结构分析是指"结构分析器"或"剖析器"。它可判别所得到的表达式在句法上是否正确。如果句法是正确的，就能得到模式的完整描述，即一个剖析式或剖析树。

图 10-21 中的分析部分包括基元选择及结构推断。模式分析是为模式识别服务的。基元选择提供参考模式基元，供识别部分作为匹配模板用，以完成识别任务。

一类模式可以用一个文法来表示，在机器中只存文法就可以了。简单的文法目前可由机器来作文法推断，较复杂的实用的文法尚需人和机器配合来推断。

基元选择和结构推断是相互关联的，基元选择得复杂一些，句法结构就可简单些；如果基元选择得很简单，文法就比较复杂，所以两者之间要折衷考虑。

根据模式的不同，模式结构的表示方法也有所不同。一维模式大都用一维链来描述。对于二维的模式，关系变得复杂了。所以，句法模式识别推广到多维时，形式语言就不适用了，需要加以推广，以适应识别的要求。

关于模式结构的表示方法，一维图形用链码串，多维模式则用树图结构，这些在第 8 章已有详细讨论，在此不再赘述。

句法方法在以下领域多有应用：

1）波形分析；

2）声音识别与理解；

3）文字识别；

4）二维数学表示式；

5）指纹分类；

6）图像分析与理解；

7）机器部件识别；

8）自动视觉检查；

9）LANDSAT 资源勘探用陆地卫星数据解释等。

10.3.3 误差校正句法分析

1. 噪声和干扰问题

在模式识别中，经常存在噪声与干扰。在句法结构法的讨论中没有考虑噪声和干扰问题。决策理论法较容易解决噪声和干扰问题，但只能作分类，不能给出描绘，也不知道模式结构。因此，在类别很多时，只作分类没有意义。句法方法能给出结构，可重新构造模式。但在有噪声时，可能会认错基元。因此，在句法结构法中如何解决噪声干扰问题是一个重要课题。对解决这样的问题，曾有各种考虑：

1）在推断文法时就考虑噪声和干扰样品；

2）在预处理中尽可能去掉噪声；

3）用随机文法，借用统计方法给句子出现加上概率分布，但推断这种文法更困难；

4）采用转换文法；

5）采用误差校正剖析；

6）把决策理论和句法结构结合起来使用。

2. 随机文法

为解决噪声干扰问题可采用随机文法进行识别。随机文法是一个四元式，它与一般文法不同之点在于每一个产生式与一定概率相联系，四元式为如下形式：

$$G_S=(V_N,V_T,P_S,S)$$

其中：V_N 为非终止符有限集；V_T 为终止符有限集；S 为起始符，$S\in V_N$；P_S 为随机产生式有限集。

产生式形式如下：

$$\alpha_i \xrightarrow{P_{ij}} \beta_{ij} \qquad j=1,2,\cdots,n \quad i=1,2,\cdots,K$$

其中 $\alpha_i\in(V_N\cup V_T)^*V_N(V_N\cup V_T)^*$，$\beta_{ij}\in(V_N\cup V_T)^*$，$P_{ij}$是与使用这个随机产生式相联系的概率，

$$0<P_{ij}\leqslant 1 \qquad \sum_{j=1}^{n}P_{ij}=1 \tag{10-46}$$

假定 $\alpha_i \xrightarrow{P_{ij}} \beta_{ij}$是在 P_S 中，则链 $\xi=r_1\alpha_i r_2$ 能以概率 P_{ij}被 $\eta=r_1\beta_{ij}r_2$ 所代换。把这个导出式表示为

$$\xi \xrightarrow{P_{ij}} \eta$$

并且可以认为 ξ 以概率 P_{ij}直接产生 η。

如果存在一个链的序列 $w_1,w_2,\cdots,w_{n+1}$，使得

$$\xi_1=w_1 \quad \eta=w_{n-1} \quad w_i\overset{P_{ij}}{\Rightarrow}w_{i-1} \qquad i=1,2,\cdots,n$$

则我们说 ξ 以概率 $P=\prod_{i=1}^{m}P_i$ 产生 η。

在有噪声的情况下，标准模式出现的概率仍然较大。既然一个模式用一个句子表示，那么一类具有确定概率分布的模式可用一组句子表示，每个句子有相应的$[x,P(x)]$，这组句子称为随机语言，它可以由一个随机文法G_S来表征。通常记作$L(G_S)$。

当有m类模式时，就有m个文法$G_{S1},G_{S2},\cdots,G_{Sm}$来表征这$m$类模式，此时$m$有个条件概率$P(x|G_{S1}),P(x|G_{S2}),\cdots,P(x|G_{Sm})$，当输入一个未知句子$y$时，就可根据最大似然办法来决定$y$属于哪一类。其原理框图如图10-22所示。这实际上就是一个最大似然分类器。

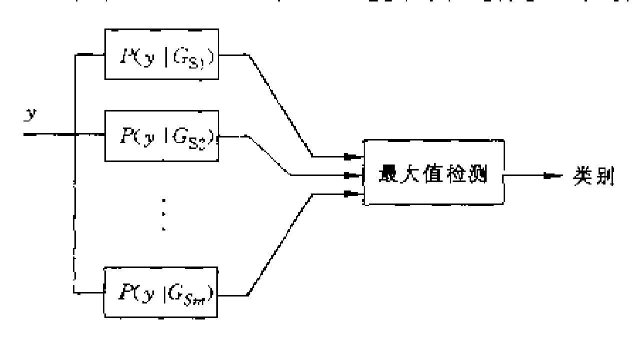


图10-22 随机文法最大似然分类器

因为随机文法产生式除了指派的概率以外，实际上和非随机文法一样，所以，一个随机文法所产生的语言的集合与非随机方式产生的相同。定义$\overline{G}_S$为表征文法，它是由随机文法G_S中去掉每个随机产生式相关联的概率而得到的。于是，

$$\overline{G}_S=(V_N,V_T,P,S)$$

它显然是非随机的文法。基于表征文法的概念，且仅当$\overline{G}_S$是0型、1型、2型、3型文法时，把随机文法G_S称为0型（无限制），1型（上下文有关），2型（上下文无关），3型（正规）随机文法。

如果L_S是一个随机正规语言，则它就是一个随机上下文无关语言。如果L_S是一个不包含λ的上下文无关语言，则它就是一个随机上下文敏感语言。如果L_S是一个随机上下文敏感语言，则它就是一个随机无限制语言。

随机文法在一定程度上能解决噪声干扰问题，但这方面的困难在于各产生式的概率不易确定。

在随机文法中，如果G_S满足条件

$$\sum_{x\in L(G_S)} P(x)=1 \tag{10-47}$$

则G_S被称为一致性文法。对于随机文法的一致性了解如下：对于随机有限状态文法，用检验马尔科夫链的方法去检验一致性条件；对随机上、下文无关文法用检验多重类型分支过程的方法检验一致性条件；对随机上下文有关文法如何检验一致性条件尚没有解决；对于随机正规文法一致性条件是显而易见的，因为每个随机3型文法都是一致性的。

3. 随机有限自动机

对于一个随机正规文法产生的语言，可以设计一个随机有限自动机来接受该语言。一个随机有限自动机是一个五元式：

$$\boldsymbol{M}_S=(\boldsymbol{\Sigma},\boldsymbol{Q},\boldsymbol{M},\boldsymbol{\Pi}_0,\boldsymbol{F})$$

其中：$\boldsymbol{\Sigma}$为输入符号集；$\boldsymbol{Q}$为内部状态有限集；$\boldsymbol{M}$为$\boldsymbol{\Sigma}$到$n\times n$随机状态转移矩阵的映射（n是$\boldsymbol{Q}$中状态数目）；Π_0为初始状态分布，它是一个n维向量；$\boldsymbol{F}$为终止状态有限集。

对于给定的随机有限状态文法 G_S 有可以构成一个随机有限自动机。其步骤如下：

设
$$G_S=(V_N,V_T,P_S,S)$$
$$\boldsymbol{M}_S=(\boldsymbol{\Sigma},\boldsymbol{Q},\boldsymbol{M},\Pi_0,\boldsymbol{F})$$

1) $\boldsymbol{M}_S$ 中的 $\boldsymbol{\Sigma}$ 等于 V_T，即 $\boldsymbol{\Sigma}=V_T$；

2) $\boldsymbol{M}_S$ 状态集 $\boldsymbol{Q}$ 等于非终止符集 V_N 和两个附加状态 T、R 的并集，T、R 分别对应于终止和拒绝状态，拒绝状态主要是为了规范化；

3) Π_0 是行向量，在 S 状态位置上，分量等于 1，其他位置等于 0；

4) 最终状态集只有一个元素，那就是 T；

5)状态转移矩阵 $\boldsymbol{M}$ 是在文法的随机产生式 P_S 的基础上形成的。如果 P_S 中有产生式 $A_i\xrightarrow{P_{ij}^l}a_lA_j$，则对于状态转移矩阵 $\boldsymbol{M}(a_l)$ 第 i 行第 j 列的元素就是 P_{ij}^l。

用随机文法解决噪声和干扰问题的缺点是为了推断随机文法和与每条产生式相联系的概率需要大量的样品，有时同时推断出文法和概率值是很困难的。

4. 误差校正句法分析

基于用随机文法解决噪声和干扰问题的缺点，考虑在文法演变过程中去消除噪声及干扰。这就是所谓转换文法。通常把经过误差校正后的扩大文法看成是一种转换文法，它是把原来没有考虑噪声的文法变成有噪声的文法，用来解决噪声与干扰问题，这种方法称为误差校正句法分析。

在有噪声的情况下，可能产生分割误差和基元识别误差。按结构分可有三种类型：

1) 代换误差

$$w_1aw_2\left|\frac{T_S}{}w_1bw_2\right. \qquad a,b\in V_T \quad a\neq b$$

2) 插入误差

$$w_1aw_2\left|\frac{T_I}{}w_1aw_2\right. \qquad a\in V_T$$

3) 抹去误差

$$w_1aw_2\left|\frac{T_D}{}w_1w_2\right. \qquad a\in V_T$$

定义两条链的距离如下：

两条链 $x,y\in V_T^*$ 之间的距离 $d^L(x,y)$ 定义为从 x 导出 y 所需的最小转换数目。有时称为利汶施坦距离。

例如：给出一个句子 $x=\text{cbabdbb}$，另一个句子 $y=\text{cbbabbdb}$。于是有

$$x=\text{cbabdbb}\left|\frac{T_S}{}\text{cbabbbb}\right|\frac{T_S}{}\text{cbabbdb}\left|\frac{T_I}{}\text{cbbabbdb}\right.=y$$

从 x 转换到 y 所需的最少转换为 3，所以

$$d^L(x,y)=3$$

如果把非负数 σ,γ,δ 分别加到转换 T_S,T_D,T_I 上，就可以定义一种加权的利汶施坦距离。

设 $x,y\in V_T^*$ 是两条链，J 是从 x 导出 y 所用到的转换序列，则 x 和 y 之间的加权利汶施坦距离，表示成如下形式：

$$d^W(x,y)=\min_J\{\sigma\cdot k_J+r\cdot m_J+\delta\cdot n_J\} \tag{10-48}$$

其中 k_J,m_J,n_J 分别表示 J 中的代换、抹去、插入转换的数目。

一种加权度量反映了存在于不同终止符上的同一类误差的差别。在一条链 w_1aw_2 中，$a\in V_T, w_1, w_2\in V_T^*$，与终止符 a 的误差转换相关联的加权量定义如下：

1) $w_1aw_2 \left| \xrightarrow{T_S, S(a,b)} w_1bw_2\right.$，$b\in V_T$，$b\neq a$，其中 $S(a,b)$ 是用 b 代换 a 的代价，设 $S(a,a)=0$。

2) $w_1aw_2 \left| \xrightarrow{T_D, D(a)} w_1w_2\right.$，其中 $D(a)$ 是从 w_1aw_2 中除去 a 的代价。

3) $w_1aw_2 \left| \xrightarrow{T_I, I(a,b)} w_1baw_2\right.$，$b\in V_T$，其中 $I(a,b)$ 是把 b 插入到 a 前面的代价。

4) $x \left| \xrightarrow{T_I, I'(b)} xb\right.$，$b\in V_T$，其中 $I'(b)$ 是在链 x 的末尾插入 b 的代价。

设 $x,y\in V_T^*$ 是两条链，J 是从 x 导出 y 所用的转换序列，$|J|$ 为与 J 中转换相关联的加权量的总和。于是，x 和 y 之间的加权距离 $d^W(x,y)$ 可定义为下式：

$$d^W(x,y) = \min_J\{|J|\} \tag{10-49}$$

式(10-49)可用图 10-23 来加以说明。

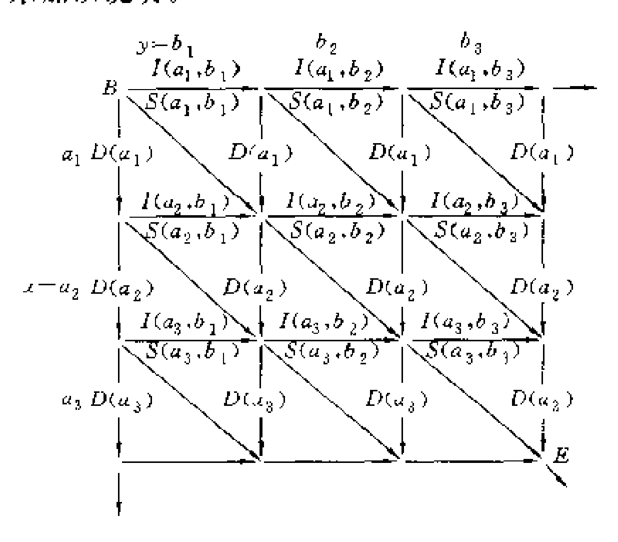


图 10-23 度量 W 的图示

图 10-23 中，从 B 到 E，网络中每一条路径对应于从 x 导出 y 所需的转换序列。水平分支表示一个插入转换，垂直分支表示抹去转换，对角线分支表示一个代换或无误差转换。分配到 x 中某个特定符号上的特定类型转换的加权量标在与其相对应的分支上。设 J 为网络上的一条路径，则 $|J|$ 是与 J 中每一分支有关联的加权量的总和。距离 $d^W(x,y)$ 是与最小加权路径相关联的总加权量。

5. 句法结构的分类与聚合的基本概念

(1) 最小距离分类

最小距离分类是研究句子之间的距离，根据距离大小来分类。

假设有 m 个模式类 $w_1, w_2, \cdots, w_m$。每一类有一个模板链，共有 m 个模板 $x_1, x_2, \cdots, x_m$。当输入一个链 y 时，则可计算利汶施坦距离 $d(x_i,y)$ 或加权利汶施坦距离 $d^W(x_i,y)$，$i=1,2,\cdots,m$。如果

$$d(x_i,y) < d(x_j,y) \qquad i\neq j \tag{10-50}$$

则指定 y 为 w_i 类。

(2) 最近邻域分类

对于 m 个模式类 $w_1, w_2, \cdots, w_m$ 每一类中有若干个模板链：

$$x_i = \{x_i^1, x_i^2, \cdots, x_i^n\} \qquad i = 1,2,\cdots,m$$

计算 $d(x_i^k, y)$ 或 $d^W(x_i^k, y)$，$i=1,2,\cdots,n$，如果：

$$\min_k \{d(x_i^k, y)\} < \min\{d(x_i^l, y)\} \tag{10-51}$$

则指定 y 为 w_i 类。

(3) 句子到句子的聚合

如果输人是一组样本 $x=\{x_1, x_2, \cdots, x_n\}$，输出要 x 分成 m 个聚合类，可按下述步骤进行：

1) 令 $j=1$，$m=1$ 指定 x_j 到 C_m 类。

2) 将 j 增加 1，对所有的 i 计算

$$D_i = \min\{d(x_i^l, x_j)\} \qquad 1 \leqslant i \leqslant m$$

令 $D_k=\min\{D_i\}$，如果 $D_k \leqslant t$，则指定 x_j 到 C_k 类；$D_k > t$，则对 x_j 建立了一个新类别，m 再增加 1。这里 t 是某一阈值。

3) 重复第二步，直到 x 中每一样本都被指定到一类为止。

在利用距离的概念时，需要在一类中找到一个或少数几个基本样板作参考，如果这一要求作不到，就需要很多样板作参考，这就要存储许多句子，此时就不如用文法推断了。

6. 误差校正剖析

(1) 最小距离误差较正剖析程序

这是一种算法。设 $L(G)$ 是一个给定的语言，y 是给定的句子。最小距离误差校正剖析的实质是在 $L(G)$ 中寻找一个句子 x，使它满足下述的最小距离准则：

$$d(x,y) = \min\{d(z,y) \mid z \in L(G)\} \tag{10-52}$$

在这里 x 称为 y 的误差校正链。

例如：对一个上下文无关语言的误差校正剖析可采用如下方法。如果有一个语言 $L(G)$，现在有一条链 y，y 有两种可能性，一是 y 在 $L(G)$ 中，另外是 y 在 $L(G)$ 外面。第一种情况可能是由于噪声的原因使 y 在 $L(G)$ 内部换成了另一个句子。第二种情况是原来的文法就不能产生 y。在第二种情况下，可将原来的文法扩大，用改变产生式的方法将三种误差加进去，使扩大了的文法产生的语言包括受噪声影响的链，然后用扩大了的文法 G' 去剖析 y，y 可以被 G' 接受，而且可以知道最少用了几次误差转换。最后再把所用到的误差转换式去掉，就可得到 $L(G)$ 中的链 x，x 就是 y 的误差校正链。

扩大的文法可用如下方法构成：

假如给定一个上下文无关文法

$$G = (V_N, V_T, P, S)$$

扩大的文法为

$$G' = (V'_N, V'_T, P', S')$$

则：1) $V'_N = V_N \cup \{S'\} \cup \{E_a \mid a \in V_T\}$，$V'_T \leqslant V_T$。

2) 如果 $A \to \alpha_0 b_1 \alpha_1 b_2 \cdots b_m \alpha_m$，$m \geqslant 0$ 是 P 中的一个产生式，$\alpha_i \in V_N^*$，$b_i \in V_T$。于是把 $A \to \alpha_0 E_{b1} \alpha_1 E_{b2} \cdots E_{bm} \alpha_m$ 合到 P' 中去，其中 $E_{bi} \in V'_N$ 是一个新的非终止符。

3) 在 P' 中加入下列产生式：

$$S' \to S$$

$$S' \to S'a \qquad a \in V_T$$

$$E_a \to a \qquad a \in V_T$$

$$E_a \to b \qquad b \neq a \in V_T$$

$$E_a \to \lambda$$

$$E_a \to bE_a \qquad b \in V_T$$

在求链与语言之间的距离过程中，结构分析用扩大的文法 G' 设计。

(2) 最近邻域结构识别

如果有两个模式类 w_1 和 w_2，分别用文法 G_1 和 G_2 加以描述。输入链 y，如果

$$d[L(G_1),y] < d[L(G_2),y] \tag{10-53}$$

则判定 y 是 w_1 类。如果

$$d[L(G_2),y] < d[L(G_1),y] \tag{10-54}$$

则判定 y 属于 w_2 类。

若有 m 类，就应把每类文法都扩大，然后进行剖析。记下每类文法剖析所用的误差转换次数，这就表示输入 y 和每类之间的距离。最后检测与某类语言的距离最小，则 y 属于该类。除了决定 y 属于哪一类外，还可以知道 y 的误差校正链 x。

这种结构识别，首先要知道各类文法，有了原来的文法 G 去产生扩大的文法 G'，再进行剖析，所以叫 ECP(Error Correcting Parsing)。

分类器框图如图 10-24 所示。

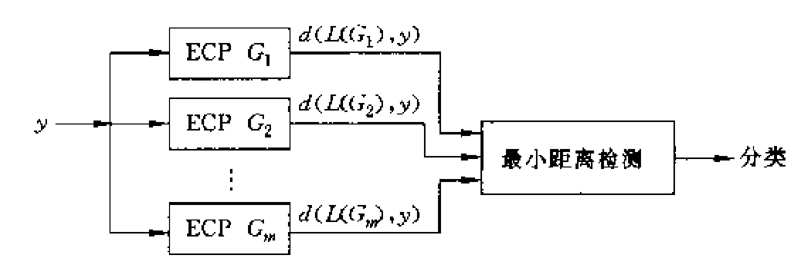


图 10-24　链到语言最小距离分类器

(3) 句法模式聚合步骤

如果有 n 个样本 $x_1,x_2,\cdots,x_n$，首先对 x_1 推断一个文法 $G_1^{(1)}$，然后将 x_2 和 $G_1^{(1)}$ 作误差校正剖析，计算 x_2 和 $L(G_1^{(1)})$ 的距离，如果距离较小，则将 x_1 和 x_2 分为一类，重新推断包括 x_1 和 x_2 的文法 $G_2^{(1)}$；如果距离较大，则将 x_2 另立一类，并推断第二类文法 $G_1^{(2)}$。这样对样本依次作下去，直到每个样本都被分到某一类中去。每一类中，每次有新的链加进来，就重新推断这一类文法。结果，n 个样本被分到 m 类中，并形成 m 类文法。综上所述，聚合步骤可归纳如下：

1) 从 x_1 推断文法 $G_1^{(1)}$；

2) 为 $G_1^{(1)}$ 构造一个误差校正剖析 $A_1^{(1)}$；

3) 用 $A_1^{(1)}$ 确定 x_2 是否与 x_1 相似，如果 x_1 和 x_2 相似，则归入一类且从 $\{x_1,x_2\}$ 推断文法 $G_2^{(1)}$，如果 x_1 和 x_2 不相似，则从 x_2 推断文法 $G_1^{(2)}$；

4)重复步骤(2)，为 $G_2^{(1)}$ 构造 $A_2^{(1)}$ 或为 $G_1^{(2)}$ 构造 $A_1^{(2)}$；

5)对新样本重复步骤(3)，直到所有样本都考虑到为止。

(4) 随机误差校正剖析

随机误差校正剖析是在误差产生式增加了概率值的剖析方法。按照误差转换的符号，与 T_S、T_I、T_D 三类转换相关联的畸变概率可定义如下：

1) 代替误差

$$w_1aw_2 \left| \frac{T_S, q_S(b|a)}{} w_1bw_2 \right.$$

其中，$q_S(b|a)$是用 b 代换 a，且 $a \neq b$ 的概率。

2）插入误差

$$w_1aw_2 \left| \frac{T_I, q_I(b|a)}{} w_1baw_2 \right.$$

其中 $q_I(b|a)$是在 a 前边插入终止符 b 的概率。

3）抹去误差

$$w_1aw_2 \left| \frac{T_D, q_D(a)}{} w_1w_2 \right.$$

其中 $q_D(a)$是抹去 a 的概率。

4）在一条链的末尾插入误差

$$x \left| \frac{T_I, q'_I(a)}{} xa \right.$$

其中，$q'_I(a)$是在一条链的末端插入终止符 a 的概率。

令 $q_S(a|a)$是 a 不发生误差的概率，可以认为它是在 a 上发生无误差转换的概率。

假定对于每一个终止符最多只能有一个误差存在，如果

$$\sum_{b \in V_T} q_S(b|a) + q_D(a) + \sum_{b \in V_T} q_I(b|a) = 1 \tag{10-55}$$

$$b \neq a \text{ 且对于所有的 } a \in V_T$$

则这种单误差模型上的畸变概率是一致的。

下面可以通过一个例子来说明如何从一条链 $x=a_1a_2a_3$ 畸变为另一条链 $y=b_1b_2b_3b_4$。畸变过程可由图 10-25 来说明。在图中，水平分支表示插入转换，垂直分支表示抹去转换，对角线分

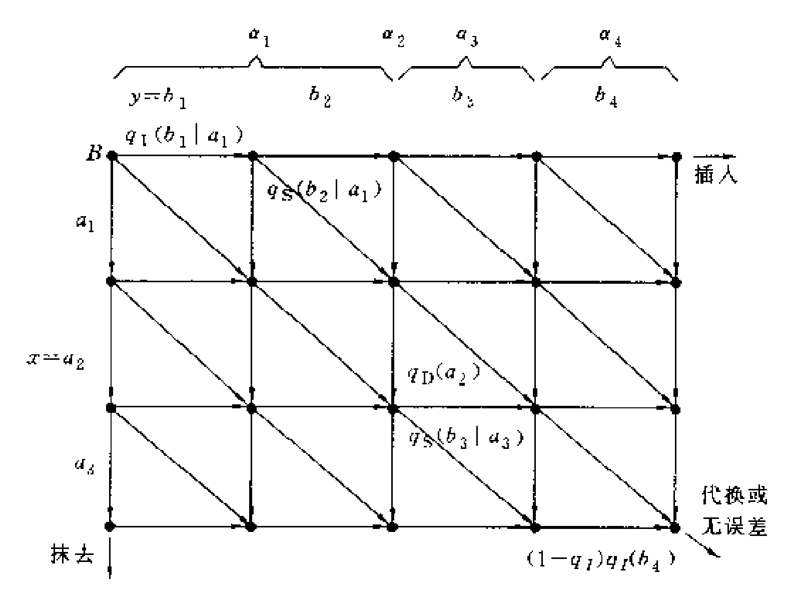


图 10-25　用网络描述的随机畸变模型

支表示代换转换或无误差转换。图中，从点 B 到点 E 的每一条通路代表 x 畸变为 y 的一种方式。粗线表示的是如下通路，b_1 是 a_1 前的一个插入，b_2 代换 a_1，a_2 被抹去，b_3 代换 a_3，b_4 插入到链尾。如果把 y 分成 $\alpha_1\alpha_2\alpha_3\alpha_4$，就可形成这种畸变，其中 $\alpha_1=b_1b_2$，$\alpha_2=\lambda$，$\alpha_3=b_3$，$\alpha_4=b_4$。这样就有：

$$q(\alpha_1\alpha_2\alpha_3\alpha_4|a_1a_2a_3)$$
$$= q_{\mathrm{I}}(b_1|a_1)q_{\mathrm{S}}(b_1|a_1)q_{\mathrm{D}}(a_2)q_{\mathrm{S}}(b_3|a_3)(1-q'_{\mathrm{I}})q'_{\mathrm{I}}(b_4) \tag{10-56}$$

当然，x 畸变到 y 可以有很多路径，可以说畸变概率 $q(y|x)$ 是从 B 点到 E 点可能性最大的通路相联系的概率。

对于 $a\in V_{\mathrm{T}}$，$\alpha\in V_{\mathrm{T}}^*$，符号 a 号畸变 α 的概率可按下式计算(畸变概率表示为 $q(\alpha|a)$)：

$$q(\alpha\ a) = \begin{cases} q_{\mathrm{D}}(a) & \alpha=\lambda \\ \max\{q_{\mathrm{S}}(b|a),q_{\mathrm{I}}(b|a)q_{\mathrm{D}}(a)\} & \alpha=b \\ q_{\mathrm{I}}(b_1|a)\cdots q_{\mathrm{I}}(b_{L-1}|a)\max\{q_{\mathrm{S}}(b_L|a),q_{\mathrm{I}}(b_L|a)q_{\mathrm{D}}(a)\} & \alpha=b_1b_2\cdots b_l,\ l>1 \end{cases} \tag{10-57}$$

将 $\alpha(\alpha\in V_{\mathrm{T}}^*)$ 插入到一条链末尾的概率为

$$q'(a) = \begin{cases} 1-q'_{\mathrm{I}} & a=\lambda \\ (1-q'_{\mathrm{I}})q'_{\mathrm{I}}(b_1)q'_{\mathrm{I}}(b_2)\cdots q'_{\mathrm{I}}(b_l)a=b_1b_2\cdots b_l & l\geqslant 1 \end{cases} \tag{10-58}$$

其中
$$q'=\sum_{a\in V_{\mathrm{T}}} q'_{\mathrm{I}}(a)$$

假定
$$q(\alpha_1\alpha_2\cdots\alpha_{n+1}|a_1a_2\cdots a_n)=q(\alpha_1|a_1)\cdots q(\alpha_n|a_n)q(\alpha_{n+1})$$

其中 $a_1,a_2,\cdots,a_n\in V_{\mathrm{T}}$，$\alpha_1,\alpha_2,\cdots,\alpha_n,\alpha_{n+1}\in V_{\mathrm{T}}^*$，则对于 x 畸变为 y 的概率如下：

$$q(y|x) = \max_i\left\{\left[\sum_{j=1}^{n}q(\alpha'_j|a_j)\right]q'(\alpha'_{n+1})\right\} \tag{10-59}$$

其中 $\alpha'_1,\alpha'_2,\cdots,\alpha'_n,\alpha'_{n+1},|\alpha'_j|\geqslant 0$，是 y 分成 $n-1$ 段子链的一种划分。

(5) 最大似然误差校正剖析

设 $L(G_{\mathrm{S}})$ 是一个随机上下文无关语言，另外有一条噪声链 y。可提出一个最大似然误差校正剖析算法，该算法在于找出一条链 x，$x\in L(G_{\mathrm{S}})$，使得

$$q(y|x)P(x) = \max_{z\in L(G_{\mathrm{S}})}\{q(y|z)P(z)\} \tag{10-60}$$

$q(y|z)$ 的计算可如前面介绍的方法进行。y 要对 $L(G_{\mathrm{S}})$ 中每个句子进行计算，然后指出畸变概率最大的情况，所以算法与 ECP 相似。

首先构成随机扩展文法。方法如下：

输入一个随机上下文无关文法：

$$G_{\mathrm{S}} = (V_{\mathrm{N}},V_{\mathrm{T}},P,S)$$

输出是一个随机扩展文法：

$$G'_{\mathrm{S}} = (V'_{\mathrm{N}},V'_{\mathrm{T}},P'_{\mathrm{S}},S')$$

其步骤如下：

1) $V'_{\mathrm{N}}=V_{\mathrm{N}}\cup\{S'\}\cup\{E_a|_{a\in V_{\mathrm{T}}}\}$；

2) $V'_{\mathrm{T}}\supseteq V_{\mathrm{T}}$；

3) 如果 $A\xrightarrow{P}a_0b_1a_1b_2\cdots b_ma_m$，$m\geqslant 0$ 是 P_{S} 中的一个产生式，a_i 在 V_{N}^* 中，b_i 在 V_{T} 中，于是在 P'_{S} 中加入产生式 $A\xrightarrow{P}a_0E_{b_1}a_1E_{b_2}\cdots E_{b_m}a_m$，$m\geqslant 0$，其中每个 E_{b_i} 都是一个新的非终止符，$E_{b_i}\in V'_{\mathrm{N}}$；

4) 在 P'_{S} 中加入下列产生式：

(a) $S\xrightarrow{1-q'_{\mathrm{I}}}S$，其中 $q'_{\mathrm{I}}=\sum_{a\in V'_{\mathrm{T}}}q'_{\mathrm{I}}(a)$；

(b) $S \xrightarrow{q_I'(a)} Sa$，对于所有的 $a \in V_T'$。

5) 对于所有的 $a \in V_T$，在 P_S'中加入下列产生式：

(a) $E_a \xrightarrow{q_S(a|a)} a$；

(b) $E_a \xrightarrow{q_S(b|a)} b$，对所有的 $b \in V_T'$，$b \neq a$；

(c) $E_a \xrightarrow{q_D(a)} \lambda$；

(d) $E_a \xrightarrow{q_I(b|a)} bE_a$，对所有的 $b \in V_T'$。

假定 y 是 x 的一个误差畸变链，$x = a_1 a_2 \cdots a_n$。用第三步中加到 P_S'的产生式，可知：

$$S \underset{G_S'}{\overset{P_i}{\Rightarrow}} X。$$

其中 $X \doteq E_{a_1} E_{a_2} \cdots E_{a_n}$，当且仅当 $S \underset{G_S}{\overset{P_i}{\Rightarrow}} x$，其中 P_i 是 G_S 中 x 的第 i 个导出式的概率。首先用步骤4)的(b)，可进一步导出 $S' \underset{G_S'}{\overset{P_i}{\Rightarrow}} Xa_{n+1}$，其中 $P \doteq P_i q(d_{n-1})$。如果 $a_1, a_2, \cdots, a_{n+1}$，是 y 的一种划分，则步骤 5)中的产生式产生，$E_{a_i} \underset{G_S'}{\xrightarrow{q(a_i|a_i)}} a_i$，对所有的 $1 \leqslant i \leqslant n$。步骤 5)中的(a)、(b)、(c)、(d)分别对应于非误差转换、代换转换、抹去转换和允许多重插入的插入转换。于是，由 G_S 产生的随机语言是：

$$L(G_S') = \{(y, P(y)) | y \in V_T^*, P(y) = \sum_{x \in G_S} \sum_{i=1}^{r} q'(y|x) P(x)\} \tag{10-61}$$

其中 r 是从 x 导出 y 的相异转换序列的数目。$q'(y|x)$是与第 i 个序列相联系的概率，$1 \leqslant i \leqslant r$。

最大似然误差校正剖析的方框图如图 10-26 所示。

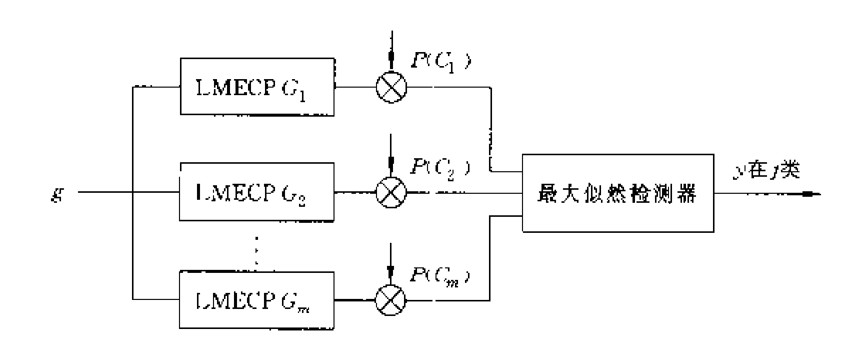


图 10-26　最大似然误差校正剖析框图

如果有两个句法模式 C_1 和 C_2，假设 x 在 C_i 中的概率为 $P(C_i)$，$i = 1, 2$ 是已知的，如果考虑贝叶斯分类，运用贝叶斯规则可知 x 在第 j 类的后验概率为

$$P(C_j|x) = \frac{P(x|C_j)P(C_j)}{\sum_{i=1}^{2} P(x|C_i)P(C_i)} \qquad i = 1, 2 \tag{10-62}$$

于是，贝叶斯决策规则为

$$x \in \begin{cases} C_1 & P(C_1|x) > P(C_2|x) \\ C_2 & P(C_1|x) < P(C_2|x) \end{cases} \tag{10-63}$$

有时候往往会出现两个文法产生同一个句子的情况，如 C_1 类产生英文字母 B，C_2 类产生

数字 8，在有噪声干扰时，B 和 8 可能在两类交叠处，也可能在两类之外。对于交叠处的可以用随机文法计算概率，用最大似然分类器解决。在两类之外可用最大似然误差校正剖析来解决。

10.3.4 文法推断

在句法模式识别中，有两个问题比较困难，一是噪声和干扰，另一个就是文法推断。文法推断是在解决了被研究模式的基元提取问题后，研究如何构成能正确描述这类模式的文法。这实际上是句法结构的学习问题。即从句子中学习文法，。由于模式可分别用链、树、图来表示，所以就有链文法推断，树文法推断以及图文法推断。

文法推断课题主要涉及推断一个未知文法 G 的句法规则所有的方法，推断的依据是 G 产生的语言 $L(G)$ 中句子或链的一个有限集合 S_T，也可能还有 $L(G)$ 的补集中的链的有限集合。推断出的文法是一种规则，它描述 $L(G)$ 中的给定有限集合，并预测给定集以外的链，这些链与给定集在某种意义上具有同样的性质。文法推断的方框图如图 10-27 所示。

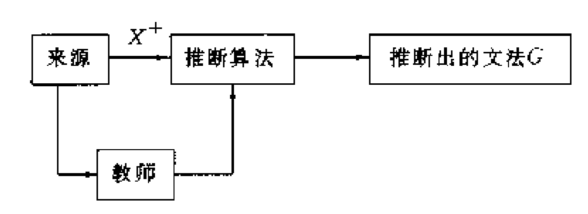


图 10-27 文法推断框图

1. 有限状态文法推断

有限状态文法可用下述方法实现：

输入 k 个样本 X^+，$X^+=\{x_1,x_2,\cdots,x_k\}$。其中 $x_i=a_{i1},a_{i2},\cdots,a_{in}$，$V_T=\{X^+$中不同的终止符$\}$。对 k 个样本中的一条链 x_i 寻找其产生式 P_i。因为正规文法的产生式只有两种形式，即 $A\to aB(A,B\in V_N)$ 及 $A\to a(a\in V_T)$，所以对于每一条链，其产生式规则为

$$
\begin{aligned}
P_i:\quad & S\to a_{i1}Z_{i1}, \qquad Z_{i1}\in V_N\\
& Z_{i1}\to a_{i2}Z_{i2}, \qquad Z_{i2}\in V_N\\
& Z_{i2}\to a_{i3}Z_{i3}, \qquad Z_{i3}\in V_N\\
& \vdots\\
& Z_{i(n-3)}\to a_{i(n-2)}Z_{i(n-2)}\\
& Z_{i(n-2)}\to a_{i(n-1)}Z_{i(n-1)}\\
& Z_{i(n-1)}\to a_{in}\\
& V_{N_i}=\{S,Z_{i1},Z_{i2},\cdots,Z_{i(n-1)}\}\\
& G_i=\{V_{N_i},V_T,P_i,S\}
\end{aligned}
$$

上述方法可通过一个例子来说明。

例 $X^+=\{01,100,111,0010\}$

$V_T=\{0,1\}$

$x_1=01$

$P_1:S\to 0Z_{11},Z_{11}\to 1$

$x_2=100$

$P_2:S\to 1Z_{21},Z_{21}\to 0Z_{22},Z_{22}\to 0$

$$x_3=111$$

$$P_3: S\to 1Z_{31}, Z_{31}\to 1Z_{32}, Z_{32}\to 1$$

$$x_4=0010$$

$$P_4: S\to 0Z_{41}, Z_{41}\to 0Z_{42}, Z_{42}\to 1Z_{43}, Z_{43}\to 0$$

$$V_N=\{S, Z_{11}, Z_{21}, Z_{22}, Z_{31}, Z_{32}, Z_{41}, Z_{42}, Z_{43}\}$$

所以这个文法共有 9 个非终止符，2 个终止符，12 个产生式。

2. 上下文无关文法的推断

在实用中，用有限数目的句子来推断文法一般总能用正规文法产生这些句子，但是有时 V_N 会很大。如果采用上下文无关文法来推断，V_N 的数目会大大减小，因此，实现起来自动机的状态也会大大减少。

上下文无关文法的推断大致有两种方法，一种方法是 Pumping Lemma 方法，另一种方法是采用具有结构性样本的推断法。

上下文无关的语言有一个特性，即如果一个句子可以分成 5 段 UVWXY，这个句子在 X^+ 内，那么形如 UV^kWX^kY 的句子也在 X^+ 内(UVWXY 定理)。例如 $aba\in X^+$，用人机对话的方式，逐个询问机器 a,b,aa,ab,ba 是否在 X^+ 内，如果机器回答 $b\in X^+$，则检验 $a^2ba^2, a^3ba^3,\cdots, a^kba^k$ 是否在 X^+ 内，如果它们都在 X^+ 内，则得到产生式：

$$S\to aSa$$

$$S\to b$$

这种方法的缺点是需要大量的外界信息，X^+ 中的句子也较多。

采用具有结构特性的样本的方法是不仅要知道句子，还要知道句子的结构。为说明方便起见，在此结构用括弧来表示。

例如：$x_1=a+a+a\in X^+$，结构信息为[[a]+[a]+[a]]，结构树如图 10-28 所示。产生这

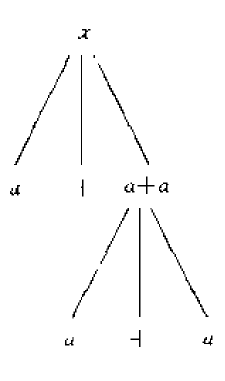


图 10-28　结构树

个句子的文法 G_1 的产生式为

$$N_1\to a$$

$$N_2\to N_1+N_1$$

$$N_2\to N_1+N_2$$

$$S\to N_2$$

$$L(G_1)=\{a+a, a+a+a, a+a+a+a,\cdots\}$$

$x_2=(a+a)+a$，它的结构信息为

$$[[([[a]+[a]])]+[a]]$$

求出产生 X_2 的文法 G_2 的产生式：

$$N_1 \to a$$
$$N_2 \to N_1 + N_2$$
$$N_3 \to (N_2)$$
$$N_4 \to N_3 + N_1$$
$$S \to N_4$$
$$L(G) = \{(a+a)+a,(a+a+a)+a,\cdots\}$$
$$L(G_1 \cup G_2) \geqslant L(G_1) \cup L(G_2)$$

这种方法需要知道句子的结构，这对于模式识别来说是可以做到的。用这个方法推断出的文法，由于基本模式数目很少，可能得到的文法质量不高，但这个缺点可用 ECP 来补偿。

3. 图文法推断

定义正样本集 $X^+=\{x_1,x_2,\cdots,x_h\}$ 和负样本集 $X^-=\{x_1,x_2,\cdots,x_h\}$，而 $X=\{x_1,x_2,\cdots,x_n\}=\{X^+,X^-\}$。

如果 X^+ 和 X^- 分别是图 10-29 所示的景物，希望有个文法产生这个图形。此图的关系图如图 10-30 所示，图文法规则如图 10-31 所示。

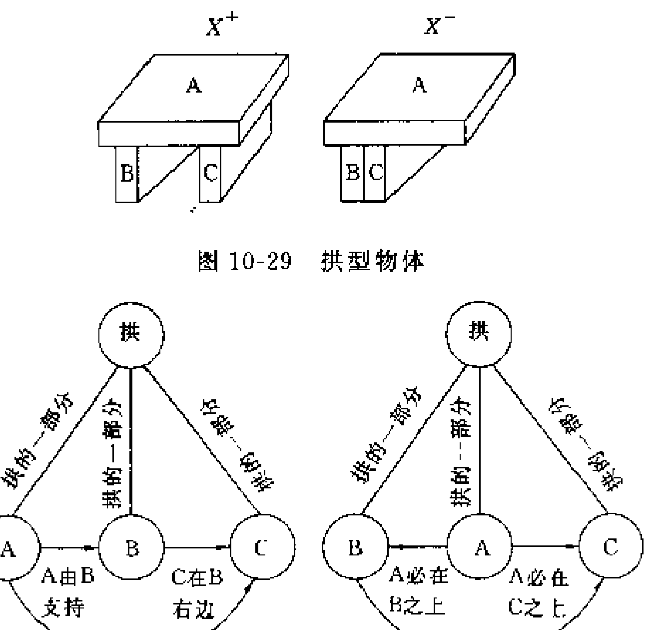


图 10-29 拱型物体

图 10-30 关系图

图文法规则如下：

第 1 个样本进来，只有 A 在 B、C 上，B 在 C 的左边；第 2 个样本进来，B 与 C 不能靠得太近；第 3 个样本进来，A 一定在 B、C 上而不能在下面；第 4 个样本……等等。按上述步骤，从根向下不断加入新的关系就可推断出图文法。

4. 树文法推断

链可以看成是只有一枝的树，而树是多枝的链。如果只对于树支全从根开始的这种特殊形状的树，则可直接把链的方法搬过来用。

5. 上下文敏感文法推断

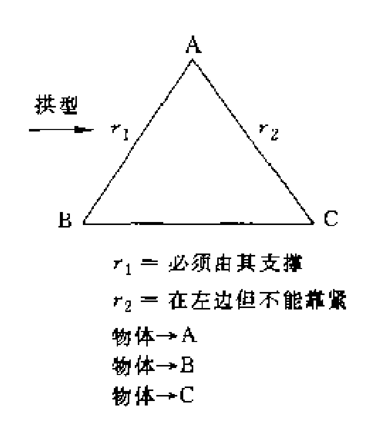


图 10-31 图文法规则

对于上下文敏感的文法(即上下文有关的文法)还没有推断办法。对于这类问题可用两种办法解决,一种是上下文无关程序文法,另一种是用属性文法。

6. 关于属性文法

在属性文法中,每一个基元都由两部分组成,基元为(b,X_b)。其中 b 表示名字, X_b 表示属性或语义信息,这是识别基元的根据。在使用产生式时,要考虑属性之间的关系。

例如:图 10-32 的树图,子模式为(B,X_B), B 的属性由 a,b,c 的属性得到。$X_B=\phi(X_a,X_b,X_c)$。式中的 ϕ 可能是简单的函数,也可能是复杂的运算式。上例中的产生式规则为 $B\to abc$,属性规则为 $X_B=\phi(X_a,X_b,X_c)$。

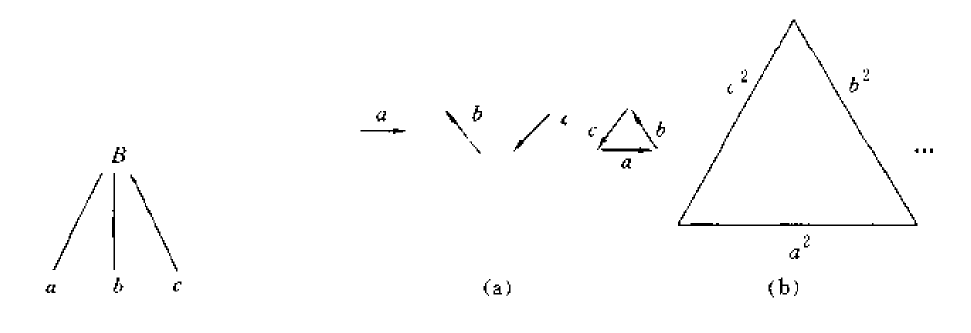


图 10-32 树图

图 10-33 基元及其所描述的图形

二者分开。属性往往是识别基元时所需要的特征量。属性并不限于一个量,可以是一组参量。属性的计算顺序是从基元属性计算起,然后是子模式属性,再往上计算模式属性。

例如 $L=\{a^nb^nc^n|_{n=1,2,\cdots}\}$是上下文无关语言,基元如图 10-33(a)所示,其所描述的图形如图 10-33 (b)所示的两个三角形。用上下文无关文法描述如下:

$$G=(V_N,V_T,P,S)$$

$$V_N=\{A,B,C\}$$

$$V_T=\{a,b,c\}$$

$$S=A$$

$$P:①A\to aBC$$

$$②B\to aBB$$

$$③C\to CC$$

$$④B\to b$$

$$⑤ C \rightarrow c$$

产生 $a^2b^2c^2$ 的过程如下：

$$A \xrightarrow{①} aBC \xrightarrow{②} aaBBC \xrightarrow{③} aaBBCC$$
$$\xrightarrow{④} aabBCC \xrightarrow{④} aabbCC \xrightarrow{⑤} aabbcC \xrightarrow{⑤} aabbcc \rightarrow a^2b^2c^2$$

如果用属性文法来描述：令 $a=a(l)$，$b=b(l)$，$c=c(l)$，其中 l 代表长度（属性），则 $L(G)=\{a(l),b(l),c(l)\}$。由这个例子可见，基元简单，文法复杂，基元复杂，文法则简单。所以基元选择很重要，应针对具体问题选择合适的方案。

付京孙教授曾提出一个实际的文法推断系统，如图 10-34 所示。这个系统采用人-机交互方式使推断问题变得容易了。一个训练者通过交互式图形显示把样本 S_t 的每条链分成 n 个不同的子链。这样就把样本集 S_t 分成 n 个子链集合。假如每个子链集合代表一个具有简单结构的子模式集合，该集合可用一个简单的文法来描述，这个文法可以用已有的方法有效地推断出来。对于图型模式而言，可按照下列准则把一个模式分成子模式：

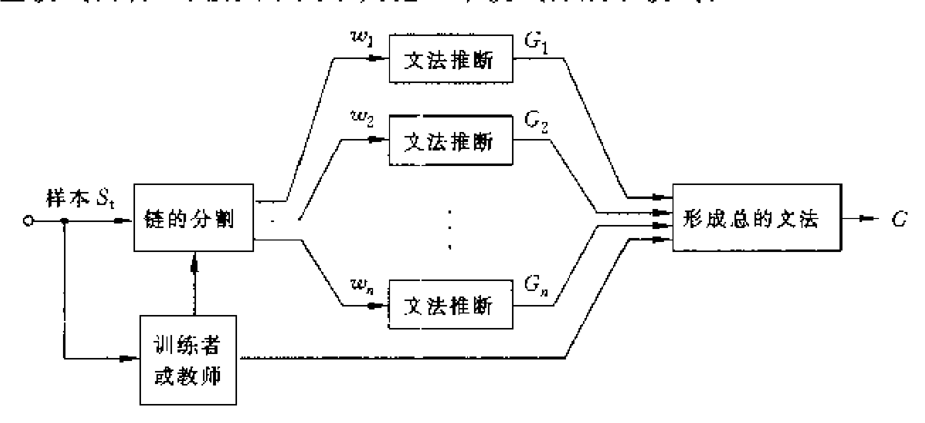


图 10-34　一个文法推断系统框图

1）选择具有简单连接结构的子模式，使每个子链集合的文法都容易推断；

2）对模式进行分割，以便充分利用重复出现的子模式。这种分割方式使得可以利用一个能接受多次出现的子链的模板文法，并且还为推断提供更多的子链样本。

把样本链集合 S_t 分割成 n 个子链集合 $w_1,w_2,\cdots,w_n$ 后，可对每个子链集合推断一个文法，这样就得到 n 个不同的文法 $G_1,G_2,\cdots,G_n$。设从 w_j 推断的文法为 $G_j=(V_{N_j},V_{T_j},P_j,S_j)$，$j=1,2,\cdots,n$。而整体文法为 $G=(V_N,V_T,P,S)$，终止符集合 V_{T_j} 属于 V_T，可能的例外是连接符 C_j 的集合，C_j 被看作是 G_j 中的终止符，G 中的非终止符，即 $C_j \in V_{T_j}$ 及 $C_j \in V_N$。

以上介绍的是文法推断的基本概念和方法，基本思路是从样本出发推断出文法。详细实施方法请参考有关专著。

10.4　模糊集识别法简介

在模式识别中，有些问题是极其复杂的，要使计算机识别某一模式，就要分析综合所有的特征，计算和比较大量的信息后才能作出判断。而人在识别过程中只根据一些模糊的印象就可以做到较准确的识别。例如在一堆照片中找一个人，只要说"找一位长脸型、皮肤白晰、高鼻梁、大眼睛的人"就可以找出来。而如果计算机识别就必须给出"面部轮廓长宽比在 1.4 以上、鼻子长 6 厘米以上、高 2.5 厘米以上、宽 3 厘米以下"等之类的数字。如果有一张照片其他条件都符

合、只是鼻高2厘米，计算机也不会认可。计算机帮助人工作时这种丝毫不肯通融的性质在模式识别中有时反而成为累赘。这主要原因是计算机是建立在二值逻辑基础上，它对事物分析的结论是"非假"即"真"。这种二值逻辑不适于处理模糊事物。在人的日常活动中，模糊概念普遍存在，例如，"暖和"、"不冷"、"较重"、"较轻"、"长点"、"短点"等等均是一些既有区别又有联系的无一定明确分界的概念。这些概念都不能用人工语言及传统的数学模型来描述。为了描述并分析自然界及人类社会中各种模糊事物，人们就要探索能表现事物模糊性的数学工具。

根据人辨识事物的思维逻辑，吸取人脑的识别特点，模糊集合论把数学从二值逻辑转向连续逻辑，这就更接近人类大脑的识别活动。由此，产生了一种相当独特的识别方法——模糊识别法。

10.4.1 模糊集合及其运算

1. 模糊子集

在自然界及人类生活中有许多概念没有明确的外延，没有明确外延的概念就称作模糊概念。模糊概念是客观事物本质属性在人们头脑中的反映，是人类社会长期发展过程中约定成俗的东西。

论域是指被讨论的全体对象，有时也称为空间，论域元素总是分明的。论域中元素从属于模糊集合的程序不是绝对的0或1，它可介于0和1之间。在模糊数学中，把元素对普通集合的绝对隶属关系加以灵活化，提出隶属度的概念。隶属度用隶属函数来描述。

(1) 隶属函数(或从属函数)

论域 $X=\{x\}$ 上的模糊集合 $\underset{\sim}{A}$ 由从属函数 $\mu_{\underset{\sim}{A}}(x)$ 来表征，其中 $\mu_{\underset{\sim}{A}}(x)$ 在实数轴闭区间[0,1]中取值，$\mu_{\underset{\sim}{A}}(x)$ 的大小反映 x 对 $\underset{\sim}{A}$ 的从属程度。任意论域 $X=\{x\}$ 上的模糊集合 $\underset{\sim}{A}$ 是指 x 中具有某种性质的元素整体，这些元素具有某个不分明的界限。对 X 中任一元素可以用[0,1]间的数来表征该元素从属于 $\underset{\sim}{A}$ 的程度。$\mu_{\underset{\sim}{A}}(x)$ 接近于1，表示 x 从属 $\underset{\sim}{A}$ 的程度很高；$\mu_{\underset{\sim}{A}}(x)$ 接近于0，说明 x 从属于 $\underset{\sim}{A}$ 的程度很低。例如：$\underset{\sim}{A}$ 若表示远大于0的实数，即 $\underset{\sim}{A}=\{x|x\gg 0\}$，则 $\underset{\sim}{A}$ 的从属函数可写成下式形式：

$$\mu_{\underset{\sim}{A}}(x)=\begin{cases}0 & x\leqslant 0\\ \dfrac{1}{1+\dfrac{100}{x^2}} & x>0\end{cases} \tag{10-64}$$

(2) 模糊集相等

设 $\underset{\sim}{A}$ 和 $\underset{\sim}{B}$ 均为 X 中的模糊集，如对 $\forall x\in X$ 均有

$$\mu_{\underset{\sim}{A}}(x)=\mu_{\underset{\sim}{B}}(x) \tag{10-65}$$

则称 $\underset{\sim}{A}$ 和 $\underset{\sim}{B}$ 相等，即：

$$\underset{\sim}{A}=\underset{\sim}{B}\Rightarrow\mu_{\underset{\sim}{A}}=\mu_{\underset{\sim}{B}}(x) \tag{10-66}$$

其中 $\forall x$ 表示所有的 x，符号 $\Leftrightarrow$ 表示等价关系。

(3) 子集

设 $\underset{\sim}{A}$ 和 $\underset{\sim}{B}$ 均为 X 中的模糊集，如果对 $\forall x\in X$ 均有

$$\mu_{\underset{\sim}{A}}(x)\leqslant\mu_{\underset{\sim}{B}}(x) \tag{10-67}$$

则称 $\underset{\sim}{B}$ 包含 $\underset{\sim}{A}$ 或称 $\underset{\sim}{A}$ 为 $\underset{\sim}{B}$ 的子集，记为 $\underset{\sim}{A}\subseteq\underset{\sim}{B}$，即

$$\underset{\sim}{A}\subseteq\underset{\sim}{B}\Leftrightarrow\mu_{\underset{\sim}{A}}(x)\leqslant\mu_{\underset{\sim}{B}}(x) \tag{10-68}$$

3

(4) 空集

$\underset{\sim}{A}$为 X 中的模糊集，如对$\forall x \in X$ 时，均有

$$\mu_{\underset{\sim}{A}}(x) = 0 \tag{10-69}$$

则 $\underset{\sim}{A}$称为空集，记为$\varnothing$，即

$$\underset{\sim}{A} = \varnothing \Leftrightarrow \mu_{\underset{\sim}{A}}(x) = 0 \tag{10-70}$$

(5) 并集

如果 $\underset{\sim}{A}$、$\underset{\sim}{B}$、$\underset{\sim}{C}$是 X 中的模糊集，如对$\forall x \in X$，有

$$\mu_{\underset{\sim}{C}}(x) = \vee \left[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right] \tag{10-71}$$

则 $\underset{\sim}{C}$就叫做 $\underset{\sim}{A}$与 $\underset{\sim}{B}$的并集，$\underset{\sim}{C}=\underset{\sim}{A} \cup \underset{\sim}{B}$即

$$\underset{\sim}{C} = \underset{\sim}{A} \cup \underset{\sim}{B} \Leftrightarrow \mu_{\underset{\sim}{C}}(x) = \vee \left[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right] \tag{10-72}$$

(6) 全集

$\underset{\sim}{A}$为 X 中模糊集，如对$\forall x \in X$ 时，均有

$$\mu_{\underset{\sim}{A}}(x) = 1 \tag{10-73}$$

则称 $\underset{\sim}{A}$为全集，记作 Ω，即

$$\underset{\sim}{A} = \Omega \Leftrightarrow \mu_{\underset{\sim}{A}}(x) = 1 \tag{10-74}$$

(7) 交集

假如 $\underset{\sim}{A}$、$\underset{\sim}{B}$、$\underset{\sim}{C}$是 X 中的模糊集，如对$\forall x \in X$ 有

$$\mu_{\underset{\sim}{C}}(x) = \wedge \left[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right] \tag{10-75}$$

那么 $\underset{\sim}{C}$就叫做 $\underset{\sim}{A}$与 $\underset{\sim}{B}$的交集，记做 $\underset{\sim}{C}=\underset{\sim}{A} \cap \underset{\sim}{B}$，即

$$\underset{\sim}{C} = \underset{\sim}{A} \cap \underset{\sim}{B} \Leftrightarrow \mu_{\underset{\sim}{C}}(x) = \wedge \left[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right] \tag{10-76}$$

其中$\wedge$表示求最小值，作前置式用时，$\wedge$可换成 min；同样并集中的$\vee$表示求最大值，作前置式用时，$\vee$可换成 max。

(8) 补集

假如 $\underset{\sim}{A}$是 X 中的模糊集，它的补集$\overline{\underset{\sim}{A}}$由下式表示：

$$\mu_{\overline{\underset{\sim}{A}}}(x) = 1 - \mu_{\underset{\sim}{A}}(x) \qquad \forall x \in X \tag{10-77}$$

$$\overline{\underset{\sim}{A}} \Leftrightarrow \mu_{\overline{\underset{\sim}{A}}}(x)$$

2. 模糊集的表示

(1) 模糊集的台

模糊集 $\underset{\sim}{A}$的台是 X 中能使 $\mu_{\underset{\sim}{A}}(x)>0$ 的元素的集合。

(2) 模糊独点集

一个模糊独点集是它的台只有一个元素的集合。可记为 $\mu_{\underset{\sim}{A}}=\mu_a/x_0$。

如 $\underset{\sim}{A}$的台仅有有限个元素 $x_1, x_2, \cdots, x_n$，且 $\mu(x_i)=\mu_i$，则

$$\mu_A = \mu_1/x_1 \cup \mu_2/x_2 \cup \cdots \cup \mu_n/x_n \tag{10-78}$$

(3) λ 水平集(λ 截集)

设 $\underset{\sim}{A}$为 $X=\{x\}$中的模糊集，则 $\underset{\sim}{A}$的水平集为

$$A_\lambda = \{x \mid \mu_{\underset{\sim}{A}}(x) \geqslant \lambda\} \tag{10-79}$$

3. 模糊集合的代数运算

(1) 代数积

模糊集合 $\underset{\sim}{A}$和 $\underset{\sim}{B}$的代数积记做 $\underset{\sim}{A} \cdot \underset{\sim}{B}$，它的从属函数为

$$\mu_{\underset{\sim}{A}\cdot \underset{\sim}{B}} = \mu_{\underset{\sim}{A}} * \mu_{\underset{\sim}{B}} \tag{10-80}$$

(2) 代数和

$\underset{\sim}{A}$和 $\underset{\sim}{B}$的代数和为 $\underset{\sim}{A}+\underset{\sim}{B}$，它的从属函数为

$$\mu_{\underset{\sim}{A}+\underset{\sim}{B}} = \begin{cases} \mu_{\underset{\sim}{A}} + \mu_{\underset{\sim}{B}} & \mu_{\underset{\sim}{A}} + \mu_{\underset{\sim}{B}} \leqslant 1 \\ 1 & \mu_{\underset{\sim}{A}} + \mu_{\underset{\sim}{B}} > 1 \end{cases} \tag{10-81}$$

(3) 环和

$\underset{\sim}{A}$和 $\underset{\sim}{B}$的环和记做 $\underset{\sim}{A}\oplus\underset{\sim}{B}$，它的从属函数为

$$\mu_{\underset{\sim}{A}\oplus\underset{\sim}{B}} = \mu_{\underset{\sim}{A}} + \mu_{\underset{\sim}{B}} - \mu_{\underset{\sim}{A}} \cdot \mu_{\underset{\sim}{B}} \tag{10-82}$$

4. 模糊集合运算的基本性质

如普通集一样，模糊集运算满足自反律、反对称律、传递律、幂等律、交换律、结合律、吸收律、分配律、复归律、对偶律。

5. 模糊熵

关于熵的概念在前边已有介绍，它是随机事件的不确定性的度量。在定义模糊熵时，也希望它具有普通熵的性质。

设 $\underset{\sim}{A}$和 $\underset{\sim}{B}$是 X 中的模糊集，则

$$d(\underset{\sim}{A},\underset{\sim}{B}) = \frac{1}{n}\sum_{i=1}^{n} \left| \mu_{\underset{\sim}{A}}(x_i) - \mu_{\underset{\sim}{B}}(x_i) \right| \tag{10-83}$$

叫做相对汉明距离。

设 $\underset{\sim}{A}$和 $\underset{\sim}{B}$是 X 中的模糊集，则

$$R(\underset{\sim}{A},\underset{\sim}{B}) = \frac{1}{\sqrt{n}} \cdot \sqrt{\sum_{i=1}^{n} (\mu_{\underset{\sim}{A}}(x_i) - \mu_{\underset{\sim}{B}}(x_i))^2} \tag{10-84}$$

称为 $\underset{\sim}{A}$,$\underset{\sim}{B}$的欧氏距离。

如果用 A°表示与模糊集有最小欧氏距离的普通集合，显然有：

$$\mu_{A^{\circ}}(x_i) = \begin{cases} 0 & \mu_{\underset{\sim}{A}}(x_i) < 0.5 \\ 1 & \mu_{\underset{\sim}{A}}(x_i) \geqslant 0.5 \end{cases}$$

令 $L(\underset{\sim}{A})=2d(\underset{\sim}{A},A^{\circ})$，则 $L(\underset{\sim}{A})$定义为模糊集 $\underset{\sim}{A}$的模糊熵，即

$$L(\underset{\sim}{A}) = \frac{2}{n}\sum_{i=1}^{n} \left| \mu_{\underset{\sim}{A}}(x_i) - \mu_{A^{\circ}}(x_i) \right| \tag{10-85}$$

模糊熵具有如下性质：

1) $L(\underset{\sim}{A})\geqslant 0$;

2) 若对任意 x，均有 $\mu_{\underset{\sim}{A}}(x)=0$ 或 $\mu_{\underset{\sim}{A}}(x)=1$，则 $L(A)=0$，也就是当 A 是普通集时，$L(\underset{\sim}{A})=0$;

3) 若对任意 x，均有 $\mu_{\underset{\sim}{A}}(x)=\frac{1}{2}$时，则 $L(\underset{\sim}{A})$达到极大，$L(\underset{\sim}{A})_{\max}=1$;

(4)若对任意 x，均有 $\mu_{\underset{\sim}{A}}(x)\geqslant\frac{1}{2}$，且 $\mu_{\underset{\sim}{B}}(x)\geqslant\mu_{\underset{\sim}{A}}(x)$或 $\mu_{\underset{\sim}{A}}(x)\leqslant\frac{1}{2}$，则当 $\mu_{\underset{\sim}{B}}(x)\leqslant\mu_{\underset{\sim}{A}}(x)$时，有 $L(\underset{\sim}{A})\geqslant L(\underset{\sim}{B})$。

10.4.2 模糊关系及性质

1. 普通关系

(1) 直积集合

设有两个集合 A 和 B,在 A 中取一个元素 x,在 B 中取一个元素 y,把它们搭配起来成为序偶 (x,y),所有这种序偶的全体构成一个集合就是直积集合,即

$$A \times B = \{(x,y) \mid x \in A, y \in B\} \tag{10-86}$$

序偶与顺序有关,也就是 $(x,y) \neq (y,x)$。例如,$A=(0,1)$,$B=(a,b,c)$,则有

$$A \times B = \{(0,a),(0,b),(0,c),(1,a),(1,b),(1,c)\}$$

$$B \times A = \{(a,0),(b,0),(c,0),(a,1),(b,1),(c,1)\}$$

(2) 关系

两个集合 A 和 B,其直积 $A \times B$ 的子集 R 称为 A 和 B 之间的二元关系,$A \times A$ 的子集称为 A 的二元关系(或 A 中的关系),R 可记做:$R \subseteq A \times B$ 或 $R \subseteq A \times A$。一般把直积 $A \times A \times \cdots \times A$ 的子集称为 A 上的 n 元关系。若 $(x,y) \in A$,$(x,y) \in R$,则 x 和 y 有关系 R,记做 xRy,若 $(x,y) \overline{\in} R$,记做 $x\overline{R}y$。关系 R 可用矩阵来表示,关系矩阵是元素仅为 0 和 1 的矩阵。

例如:$X=\{x_1,x_2,x_3\}$,$Y=\{y_1,y_2\}$,则 $R=\{(x_1y_1),(x_2y_1),(x_2y_2),(x_3y_2)\}$,关系矩阵表示如下:

$$R = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

若 $I=\{(x,x) \mid x \in X\}$,则称 I 为恒等关系,其相应矩阵为单位矩阵。若 $E=\{(x,y) \mid x,y \in X\}$,即 $\forall\ x,y \in X$ 有 xRy,则称 E 为全称关系,相应矩阵为 $M_E = E$。

(3) 关系的运算

假设 R 和 S 是 X 到 Y 的关系,即:$R \subseteq X \times Y$,$S \subseteq X \times Y$。R 和 S 相应的矩阵为 M_R 和 M_S,且 $M_R=[a_{ij}]$,$M_S=[b_{ij}]$,则关系运算就对应了矩阵的运算,运算的定义如下:

1) 相等

$$R = S \Leftrightarrow M_R = M_S \Leftrightarrow a_{ij} = b_{ij} \tag{10-87}$$

2) 包含

$$R \subseteq S \Leftrightarrow M_R \subseteq M_S \Leftrightarrow a_{ij} \leqslant b_{ij} \tag{10-88}$$

3) 并

设 $$M_{R \cup S} = [c_{ij}] \qquad c_{ij} = a_{ij} \vee b_{ij}$$

则 $$R \cup S \Leftrightarrow M_{R \cup S} = M_R \vee M_S \tag{10-89}$$

4) 交

设 $$M_{R \cap S} = [c_{ij}]$$

则 $$R \cap S \Leftrightarrow M_{R \cap S} = M_R \wedge M_S \tag{10-90}$$

其中 $$c_{ij} = a_{ij} \wedge b_{ij}$$

5) 补

设 $$M_{\overline{R}} = [c_{ij}]$$

则 $$\overline{R} \Leftrightarrow \overline{M}_R = \overline{M}_R \quad c_{ij} = \overline{a}_{ij} \tag{10-91}$$

6) 合成

若 X 到 Y 的关系为 R,Y 到 Z 的关系为 S,把 X 到 Z 的由下式定义的关系称为 R 和 S 的合成关系,记为 $R \circ S$,其相应的运算称为合成运算:

$$M_{R \circ S} = M_R \circ M_S$$

如果 $$M_{R\cdot S}=[c_{ij}]$$

则 $$c_{ij}=\bigvee_{k=1}^{n}(a_{ik}\wedge b_{kj}) \tag{10-92}$$

（4）关系性质

1）自反性和反自反性：R 是 X 里的关系。若 R 是自反关系，则以 $(\forall x)(x\in X\rightarrow xRx)$ 表示，其相应矩阵满足：

$$M_I\leqslant M_R$$

若 R 是反自反关系，则以 $(\forall x)(x\in X\rightarrow x\overline{R}y)$ 表示，其相应矩阵对角线元素皆为 0。

2）对称和反对称性：R 是对称关系，则以 $(\forall x)(\forall y)(x\in X\wedge y\in Y, xRy\rightarrow yRx)$ 表示，相应矩阵满足：

$$M_R^T=M_R \quad (M_R^T \text{是} M_R \text{的转置}) \tag{10-93}$$

R 是反对称关系，则以 $(\forall x)(\forall y)(x\in X\wedge y\in X\wedge xRy\wedge yRx\rightarrow x=y)$

$$M_R\wedge M_R^T\leqslant M_I \tag{10-94}$$

3）传递性：R 为传递关系，则以 $(\forall x)(\forall y)(\forall z)(x\in X\wedge y\in X\wedge z\in X\wedge xRy\wedge yRx\rightarrow xRz)$ 表示。相应矩阵满足

$$M_R\circ M_R\leqslant M_R \tag{10-95}$$

以上是关系的三个性质。如果 R 满足自反性和对称性就称为相容关系；如果 R 满足自反性、对称性和传递性，则称 R 为等价关系。

对于一个集，根据某种关系或某种观点把集中某些元看成相等或同类，把某些元看成不相等或不同类，叫做分类。分类所采用的关系或观点是一个等价关系，因此，对于一个集，有一个等价关系，就有一种分类；反之，若有一种分类，就有一个等价关系。

2. 模糊关系

模糊关系是普通关系的拓广。普通关系描述元素之间是否有关连，而模糊关系描述元素之间的关连是多少。

（1）模糊关系

直积空间 $X\times Y=\{(x,y)|x\in X,y\in Y\}$ 中的模糊关系 $\underset{\sim}{R}$ 是 $X\times Y$ 中的模糊集 $\underset{\sim}{R}$，$\underset{\sim}{R}$ 的从属函数用 $\mu_{\underset{\sim}{R}}(x,y)$ 表示。特殊情况下 $X\times X$ 中的模糊关系就称为 X 上的模糊关系。一般 $X=X_1\times X_2\times\cdots\times X_n$ 中的 n 项模糊关系 $\underset{\sim}{R}$ 是 X 中的模糊集 $\underset{\sim}{R}$，$\underset{\sim}{R}$ 的从属函数用 $\mu_{\underset{\sim}{R}}(x_1,x_2,\cdots,x_n)$ 表示。

（2）模糊关系的运算

1）相等

$$\underset{\sim}{R}_1=\underset{\sim}{R}_2\Leftrightarrow \mu_{\underset{\sim}{R}_1}(x,y)=\mu_{\underset{\sim}{R}_2}(x,y) \qquad \forall x,y\in X \tag{10-96}$$

2）包含

$$\underset{\sim}{R}_1\subseteq \underset{\sim}{R}_2\Leftrightarrow \mu_{\underset{\sim}{R}_1}(x,y)\leqslant \mu_{\underset{\sim}{R}_2}(x,y) \qquad \forall x,y\in X \tag{10-97}$$

3）并

$$\underset{\sim}{R}_1\cup \underset{\sim}{R}_2\Leftrightarrow \mu_{\underset{\sim}{R}_1\cup \underset{\sim}{R}_2}(x,y)=\vee[\mu_{\underset{\sim}{R}_1}(x,y),\mu_{\underset{\sim}{R}_2}(x,y)] \qquad \forall x,y\in X \tag{10-98}$$

4）交

$$\underset{\sim}{R}_1\cap \underset{\sim}{R}_2\Leftrightarrow \mu_{\underset{\sim}{R}_1\cap \underset{\sim}{R}_2}(x,y)=\wedge[\mu_{\underset{\sim}{R}_1}(x,y),\mu_{\underset{\sim}{R}_2}(x,y)] \qquad \forall x,y\in X \tag{10-99}$$

5）补

$$\overline{R} \Leftrightarrow \mu_{\overline{R}}(x,y) = 1 - \mu_{R}(x,y) \tag{10-100}$$

6)合成

$$R_1 \circ R_2 \Leftrightarrow \mu_{R_1 \circ R_2}(x,y) = \bigvee_z [\mu_{R_1}(x,z) \wedge \mu_{R_2}(z,y)] \quad \forall x,y \in X \tag{10-101}$$

若 R为 X 上的模糊关系，$X=\{x_1,x_2,\cdots,x_n\}$，则 R可表示成 n 阶方阵。模糊关系矩阵的运算基本上和普通关系运算一致，仅是在模糊运算中 $\vee$ 表示 max，$\wedge$ 表示 min。

(3) 模糊关系的性质

1) 自反性和反自反性　R是 X 中的模糊关系，对$\forall x \in X$，若有 $\mu_R(x,x)=1$ 成立，则称 R 满足自反性，相应矩阵满足：

$$M_I \leqslant M_R \tag{10-102}$$

若 $\mu_R(x,x)=0$，则称 R具有反自反性，即 R对角线元素皆为 0。

2) 对称性和反对称性　R是 X 中模糊关系，对$\forall x,y \in X \times X$，若 $\mu_R(x,y)=\mu_R(y,x)$成立，则称 R具有对称性，相应矩阵满足：

$$R^T = R \tag{10-103}$$

若 $\mu_R(x,y) \neq \mu_R(y,x) \Leftrightarrow \mu_R(x,y)=\mu_R(y,x)=0$，则称 R满足反对称性，相应矩阵满足：

$$R \cdot R^T \leqslant I \tag{10-104}$$

3) 传递性　R是 X 中模糊关系，$\forall (x,y),(y,z),(x,z) \in X \times X$，若均存在 $\mu_R(x,z) \geqslant \vee_y [\mu_R(x,y) \wedge \mu_R(y,z)]$成立，则称 R满足传递性，相应矩阵满足：

$$R \circ R \leqslant R \quad 或 \quad R^2 \leqslant R \tag{10-105}$$

若 R满足自反性和对称性，则称 R为模糊相容关系；若满足自反性、对称性和传递性，则称 R为模糊等价关系。与普通关系一样，模糊集分类所依据的关系也是模糊等价关系。

10.4.3 模糊模式识别的方法

在通常的模式识别中，模式是明确、清晰、肯定的。但也有很多实际问题，模式本身就不很明确，因此，描述这些模式最好用模糊集，对"模糊模式"可用模糊识别法来识别。

1. 隶属原则和模糊模式识别的直接方法

设 $A_1,A_2,\cdots,A_n$是论域 U 上的 n 个模糊子集，若对每一个 A_i 都建立一个从属函数$\mu_{A_i}(u)$，对于任一元素 $u_0 \in U$，若满足

$$\mu_{A_i}(u_0) = \max\{\mu_{A_1}(u_0),\mu_{A_2}(u_0),\cdots,\mu_{A_n}(u_0)\} \tag{10-106}$$

则认为 μ_i 隶属于 A_i，这就是隶属原则。

直接计算元素的从属函数来判断模式归属的方法称为模糊识别的直接方法，其识别效果依赖于模式从属函数。如何合理地确定出从属函数，至今仍无规律可循，而主要靠实际经验。下边以实例说明如何建立从属函数。

例　三角形的模糊集分类。

在模式识别中，任何复杂的图像都可以看成是由简单的几何图形组成的，研究了简单几何图形的分类及其组成规律，便可进一步识别复杂的图像。

如果给出一个三角形，如何判断它是等腰三角形、直角三角形、等腰直角三角形、等边三角形、一般三角形等，这是一个分类问题。当然，这里所说的各种三角形并不是几何中严格定义的三角形，而是在人们头脑中带有一定模糊性的概念。因此，这是一个模糊分类问题。任何三角形都可用三个边 a、b、c 及三个顶角 A、B、C 来表示。把等腰三角形、等边三角形、直角三角形、

等腰直角三角形看成是模糊集 $\underline{I}$、$\underline{E}$、$\underline{R}$、$\underline{IR}$。要运用直接方法识别，首先要确定它们的从属函数。

取论域：

$U=\{(A,B,C)\mid A>0,B>0,C>0,A+B+C=180°\}$，其中 A、B、C 表示三角形的三个内角，由此，可定义它们的从属函数，进一步求得模糊几何图形的从属度。

设 $\mu_{\underline{I}}(A,B,C)$、$\mu_{\underline{R}}(A,B,C)$、$\mu_{\underline{E}}(A,B,C)$、$\mu_{\underline{IR}}(A,B,C)$、$\mu_{\underline{T}}(A,B,C)$分别为等腰三角形、直角三角形、等边三角形、等腰直角三角形及非典型一般三角形的从属函数，则有

$$\mu_{\underline{I}}(A,B,C)=\left[1-\frac{1}{60}\min\{A-B,B-C\}\right]^2 \tag{10-107}$$

$$\mu_{\underline{R}}(A,B,C)=\left[1-\frac{1}{90}|A-90|\right]^2 \tag{10-108}$$

$$\mu_{\underline{E}}(A,B,C)=\left[1-\frac{1}{180}(A-C)\right]^2 \tag{10-109}$$

$$\mu_{\underline{IR}}(A,B,C)=\min\left\{\left[1-\frac{1}{60}\min\{A-B,B-C\}\right]^2,\left[1-\frac{1}{90}|A-90|\right]^2\right\} \tag{10-110}$$

$$\mu_{\underline{T}}(A,B,C)=\min\{[1-\mu_{\underline{I}}(A,B,C)],[1-\mu_{\underline{E}}(A,B,C)],[1-\mu_{\underline{R}}(A,B,C)]\} \tag{10-111}$$

如果有三角形甲，其内角分别为 95°、50°、35°；三角形乙，其内角分别为 120°、40°、20°，根据隶属原则能确定它们分属哪一类三角形。

$$\mu_{\underline{I}}(95,50,35)=\left[1-\frac{1}{60}\min\{95-50,50-35\}\right]^2=\left(1-\frac{15}{60}\right)^2=0.562$$

$$\mu_{\underline{R}}(95,50,35)=\left[1-\frac{1}{90}\min\{95-90\}\right]^2=\left(1-\frac{5}{90}\right)^2=0.892$$

$$\mu_{\underline{E}}(95,50,35)=\left[1-\frac{1}{180}\min\{95-35\}\right]^2=\left(1-\frac{60}{180}\right)^2=0.444$$

$$\mu_{\underline{IR}}(95,50,35)=\min\{\mu_{\underline{I}}(95,50,35),\mu_{\underline{R}}(95,50,35)\}=0.562$$

$$\mu_{\underline{T}}(95,50,35)=\min\{1-0.562,1-0.892,1-0.444\}=0.108$$

由隶属原则，判定三角形甲是直角三角形。

$$\mu_{\underline{I}}(120,40,20)=\left[1-\frac{1}{60}\min\{120-40,40-20\}\right]^2=\left(1-\frac{20}{60}\right)^2=0.444$$

$$\mu_{\underline{R}}(120,40,20)=\left[1-\frac{1}{90}\{120-90\}\right]^2=\left(1-\frac{30}{90}\right)^2=0.444$$

$$\mu_{\underline{E}}(120,40,20)=\left[1-\frac{1}{180}\{120-20\}\right]^2=\left(1-\frac{100}{180}\right)^2=0.198$$

$$\mu_{\underline{IR}}(120,40,20)=\min\{\mu_{\underline{I}}(120,40,20),\mu_{\underline{R}}(120,40,20)\}=0.444$$

$$\mu_{\underline{T}}(120,40,20)=\min\{1-0.444,1-0.444,1-0.198\}=0.556$$

由隶属原则，判定三角形乙是一般三角形。

2. 择近原则与模糊模式识别的间接方法

择近原则是根据贴近度建立起来的一种判别方法。根据择近原则识别模式的方法就是模式识别的间接方法。下面首先说明贴近度的概念。

若 $\underline{A}$与 $\underline{B}$为论域 U 上的模糊集，称由下式所规定的数为 $\underline{A}$与 $\underline{B}$的贴近度，记为$(\underline{A},\underline{B})$，即

$$(\underset{\sim}{A},\underset{\sim}{B})=\frac{1}{2}[A\otimes B+(1-\underset{\sim}{A}\odot\underset{\sim}{B})] \tag{10-112}$$

为说明贴近度，再引入下述概念：

对于论域 U 上的模糊集 $\underset{\sim}{A}$，称：

$$\bigwedge_{u\in U}\mu_{\underset{\sim}{A}}(u)=\inf_{u\in U}\mu_{\underset{\sim}{A}}(u) \tag{10-113}$$

为模糊集 $\underset{\sim}{A}$的下模，记为 $\underline{A}$，称：

$$\bigvee_{u\in U}\mu_{\underset{\sim}{A}}(u)=\sup_{u\in U}\mu_{\underset{\sim}{A}}(u) \tag{10-114}$$

为模糊集 $\underset{\sim}{A}$的上模，记为 $\overline{A}$。

对于论域 U 上的模糊集 $\underset{\sim}{A}$与 $\underset{\sim}{B}$，称 $\underset{\sim}{A}\otimes\underset{\sim}{B}$为 $\underset{\sim}{A}$与 $\underset{\sim}{B}$的内积，$\underset{\sim}{A}\odot\underset{\sim}{B}$为 $\underset{\sim}{A}$与 $\underset{\sim}{B}$的外积，即

$$\underset{\sim}{A}\otimes\underset{\sim}{B}=\bigvee_{u\in U}(\mu_{\underset{\sim}{A}}(u)\wedge\mu_{\underset{\sim}{B}}(u)) \tag{10-115}$$

$$\underset{\sim}{A}\odot\underset{\sim}{B}=\bigwedge_{u\in U}(\mu_{\underset{\sim}{A}}(u)\vee\mu_{\underset{\sim}{B}}(u)) \tag{10-116}$$

模糊集的上模与下模，内积与外积有如下性质：

$$(\underset{\sim}{A}\otimes\underset{\sim}{B})^c=\underset{\sim}{A}^c\odot\underset{\sim}{B}^c\qquad(\underset{\sim}{A}^c\text{ 为 }\underset{\sim}{A}\text{ 的补集}) \tag{10-117}$$

$$\underset{\sim}{A}\otimes\underset{\sim}{A}=\overline{A} \tag{10-118}$$

$$\underset{\sim}{A}\odot\underset{\sim}{A}=\underset{\sim}{A} \tag{10-119}$$

$$\underset{\sim}{A}\otimes\underset{\sim}{A}\leqslant\overline{A}\wedge\overline{B} \tag{10-120}$$

$$\underset{\sim}{A}\odot\underset{\sim}{B}\geqslant\underset{\sim}{A}\wedge\underset{\sim}{B} \tag{10-121}$$

若 $\underset{\sim}{B}\supseteq\underset{\sim}{A}$，则
$$\underset{\sim}{A}\otimes\underset{\sim}{B}=\overline{A} \tag{10-122}$$

若 $\underset{\sim}{B}\subseteq\underset{\sim}{A}$，则
$$\underset{\sim}{A}\odot\underset{\sim}{B}=\underset{\sim}{A} \tag{10-123}$$

例：$U=\{a,b,c,d,e,f\}$

$$\underset{\sim}{A}=\frac{0.5}{a}+\frac{0.7}{b}+\frac{1}{c}+\frac{0.9}{d}+\frac{0.6}{e}+\frac{0.3}{f}$$

$$\underset{\sim}{B}=\frac{0.7}{a}+\frac{0.8}{b}+\frac{0.9}{c}+\frac{1}{d}+\frac{0.7}{e}+\frac{0.5}{f}$$

求 $\underset{\sim}{A}\otimes\underset{\sim}{B}$及 $\underset{\sim}{A}\odot\underset{\sim}{B}$。

$$\begin{aligned}\underset{\sim}{A}\otimes\underset{\sim}{B}&=(0.5\wedge 0.7)\vee(0.7\wedge 0.8)\vee(1\wedge 0.9)\vee(0.9\wedge 1)\\&\quad\vee(0.6\wedge 0.7)\vee(0.3\wedge 0.5)\\&=0.5\vee 0.7\vee 0.9\vee 0.9\vee 0.6\vee 0.3=0.9\end{aligned}$$

$$\begin{aligned}\underset{\sim}{A}\odot\underset{\sim}{B}&=(0.5\vee 0.7)\wedge(0.7\vee 0.8)\wedge(1\vee 0.9)\wedge(0.9\vee 1)\\&\quad\wedge(0.6\vee 0.7)\wedge(0.3\vee 0.5)=0.7\wedge 0.8\wedge 1\wedge 1\wedge 0.7\wedge 0.5=0.5\end{aligned}$$

$\underset{\sim}{A}$与 $\underset{\sim}{B}$的贴近度为

$$(\underset{\sim}{A},\underset{\sim}{B})=\frac{1}{2}[0.9+(1-0.5)]=0.7$$

对于模糊集来说，最佳贴近的必要条件是 $\underset{\sim}{A}\otimes\underset{\sim}{B}$尽可能大，而 $\underset{\sim}{A}\odot\underset{\sim}{B}$尽可能小。

设 U 上有 n 个模糊子集，$\underset{\sim}{A}_1,\underset{\sim}{A}_2,\cdots,\underset{\sim}{A}_n$，若有 $i\in\{1,2,\cdots,n\}$使

$$(\underset{\sim}{B},\underset{\sim}{A}_i)=\max_{1\leqslant j\leqslant n}(\underset{\sim}{B},\underset{\sim}{A}_j) \tag{10-124}$$

则称 $\underset{\sim}{B}$与 $\underset{\sim}{A}_i$ 最贴近。

若 $\underset{\sim}{A}_1,\underset{\sim}{A}_2,\cdots,\underset{\sim}{A}_n$ 是 n 个已知模式，$\underset{\sim}{A}_i$ 满足式(10-124)，则可断言 $\underset{\sim}{B}$应归入模式 $\underset{\sim}{A}_i$，这个原理称之为择近原则。

在模糊模式识别中，有两大类型：一类是实物模型是模糊的，被识别对象是确定的，因而考

虑的是元素对模糊集的关系，一般采用隶属原则归类；另一类是不但模型是模糊的，被识对象也是模糊的，这时考虑的是模糊集与模糊集之间的关系，使用择近原则分类。

3. 模糊聚类分析

聚类分析是将所考察的模式进行合理分类的数学方法。为了确定各样本之间的关系，常常用两种量来衡量样本间的接近程度，这就是相似系数和距离。相似系数越接近于1，样本越接近，距离越小样本也越接近。相似系数有夹角余弦、相关系数等几种定义。如果用 d_{ij} 表示样本 X_i 与 X_j 样本之间的距离，则也有如下一些距离的定义：

(1) 绝对值距离

$$d_{ij}=\sum_{k=1}^{m}|x_{ik}-x_{jk}| \tag{10-125}$$

(2) 欧氏距离

$$d_{ij}=\sqrt{\sum_{k=1}^{m}(x_{ik}-x_{jk})^2} \tag{10-126}$$

(3) 马氏距离

$$d_{ij}=\sqrt{(X_i-X_j)V^{-1}(X_i-X_j)^{\mathrm{T}}} \tag{10-127}$$

式中 V 是一个 $m\times m$ 阶的协方差矩阵，其元素为

$$V_{ij}=\frac{1}{n-1}\sum_{k=1}^{m}(x_{ki}-\bar{y}_i)(x_{kj}-\bar{y}_j)$$

(4) 兰氏距离

$$d_{ij}=\sum_{k=1}^{n}\left|\frac{x_{ik}-x_{jk}}{x_{ik}+x_{jk}}\right| \tag{10-128}$$

聚类分析的基本思想是将比较接近的样本归为一类。系统聚类法可分三个步骤进行：第一，计算各样本之间距离，将距离最近的两点合并为一类；第二，定义类与类间的距离，将最近的两类合并为新的一类；第三，反复做第二步，使类与类之间不断合并，最后完成聚类分析。

类与类之间的定义有最小距离法、最大距离法、中间距离法、重心法等。

除系统聚类法外，还有所谓动态聚类法，它不同于系统聚类法的一次形成分类结果，而是首先选择聚类中心，然后进行初始分类，此后，再根据某种最优原则进行反复修改，直到分类合理为止。聚类中心的选择方法有人为选择、随机选择、重心法、密度法等等。初始聚类形成后的修改方法可采用成批修改法或逐个修改法。相比之下，在很多情况下动态聚类法相当实用。

ISODATA 分类法是一种模糊聚类分析法。它是区别于硬分类的软分类。对于有些问题，常常不应认为样本一定属于某一类而不属于其他任何类。也就是样本从某些特征考虑应属于这一类，而从另外一些特征来看又好像应属于另一类，在这种情况下可借助于模糊集理论，认为样本以某种从属程度属于这一类，而又以某种从属程度属于另一类。这样一来，每一类都是样本集上的一个模糊子集。

设一样本集 $X=\{x_1,x_2,\cdots,x_n\}$，若要将其分成 c 类，则它的每一个分类结果都对应一个矩阵 U，由于每一类都是一个模糊子集，所以分类矩阵 U 是一个模糊矩阵。模糊矩阵 U 应满足下述三个条件：

1) $u_{ij}\in[0,1]$ 即矩阵元素在0与1之间取值；

2) $\sum_{i=1}^{c}u_{ij}=1$ 即每列元素之和为1，对一个样本而言它对各类的从属度之和为1；

3) $\sum_{i=1}^{n} u_{ik} > 0$,这一条件保证了每一类不空。

由此可见,对应样本集 X 的任一种 c 组分类,都有一个模糊矩阵 U 与之对应;反之任一满足上述条件的矩阵 U 也都对应着样本集 X 上的一种 c 组软分类。

用 M_{fc}代表所有矩阵 U 的集合,则称它为 X 的 c 组软分类空间。

为了获得合理的软分类,必须遵循某种分类准则,也就是要有一个聚类准则和聚类判据。一般用下式作为聚类依据:

$$J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m \| x_k - V_i \|^2 \tag{10-129}$$

式中 V_i 代表第 i 类聚类中心,$\| x_k - V_i \|$ 表示样本 x_k 与聚类中心 V_i 的距离平方。式(10-129)的实际意义就是各类样本到该类聚类中心的距离平方和。聚类准则是求出 U,V,使式(10-129)所表示的泛函达到最小,故此准则又可写成下式:

若 $U^* \in M_{fc}, V^* \in V$,

$$J_m(U^*,V^*) = \min\{J_m(U,V)\} \tag{10-130}$$

则 U^* 即是 X 上的最佳 c 组软分类。

式(10-129)中的 m 是一个参数,m 越大,则分类越模糊,一般 $m>1$;如果 $m=1$ 就是硬分类。

以上对模糊集及模糊集识别方法作了简要的介绍。模糊集识别法是正处于发展阶段的识别方法。除了前面介绍过的几种方法外,还有基于模糊逻辑、模糊语言及模糊概率理论的识别方法,这些识别方法既有其数学基础(模糊数学)又更接近于人的思维方法。例如在二值逻辑中,是"非真即假"的一刀切的判断方法,而在自然命题中,许多事情的判断并非如此绝对,而是多带有模糊性质。例如"今天天气很好","他很胖"。等判断就是模糊判断。模糊判断有其客观标准,只是不能简单地用 0 和 1 来区分罢了。所以模糊逻辑是研究模糊命题的连续性逻辑。所谓模糊语言就是带有模糊性的语言。它包括自然语言。模糊语言和模糊推理逻辑的任务是对人类的语言和思维进行定量分析,为人类的智能寻找合适的数学模型。模糊语言包括语言集合、似然推理及模糊文法。在模糊语言中对词义、词法和句法作了定义。由于单词的序列与对象的集合之间不仅存在着一种一一对应的明确形式,而且往往有着某种模糊关系,因此,模糊语言的概念比历来的形式语言有更广泛、更一般的意义,也更加接近自然语言。利用模糊似然推理及模糊文法进行的模糊识别也更加接近人类的自然思维方法。由于人类的思维活动是一个具有大量模糊性的推理过程,并且人们总是根据需要汲取尽量少的模糊信息进行综合推理,从而得出正确判断。所以人类识别活动比任何机器都优越。由此可以设想采用连续逻辑的识别机是新型计算机的发展方向之一。模糊集识别法在图像分类及处理中将有更加广阔的研究与应用前景。

10.5 模式识别的几种应用

模式识别的应用较广,大致可有如下几个方面:字符识别;医学诊断;遥感图像解译;人脸和指纹鉴别;污染监测;自动检查和自动化;可靠性;社会经济;语音识别和理解;考古等。下面介绍一些实例。

10.5.1 指纹识别

指纹具有两大特性，第一是没有两个人的指纹是相同的；第二是当指纹不受损伤时终生不变。所以它是识别人最有力的手段之一。指纹本身是一个无穷类问题，在应用中有不同的情况。一种情况是对指纹进行核对查找，这是一个匹配问题，当然不是匹配每根隆线，而是匹配特征。如果档案数目很大，就要进行分类，把无穷类问题变成有限类问题，以减轻匹配负担。指纹分析是标准的结构分析，分成小块后只需测量隆线的斜率，通常采用0～7，共8个方向。

首先，指纹分为七类（平斗、左箕、右箕、平弓、帐弓、左双箕、右双箕），第一类再分为18个小类，然后测量斜率。总的过程是分类、分层、分窗口，在这个过程中包括细化，连接断线等处理。尔后整个窗口用一个树代表，树的每一个分支是窗口中的一根隆线，然后找出文法，最后做一树状自动机。据有关专家说，实验中大约有10%的指纹由于噪声大而难以识别。识别一个指纹大约要50秒，40秒用于前后处理，10秒用于结构分析。美国在1965年开始进行指纹识别自动化研究，在1972年完成了叫做FINGER的系统。北京大学模式识别国家实验室的指纹识别研究历经十几年的研究，已在公安部门及银行中得到了实际应用。

10.5.2 模式识别在医学上的应用

模式识别在医学图像方面的应用还不多，主要是医学图片在预处理和分割等方面的问题还没有解决，大部分工作尚在解决此类问题。

染色体分类是句法方法的一个例子，目前只用于形状分类，其实真正染色体的分类还要用到染色体本身灰度的变化。一般作法是先找到染色体，然后扫描、分开、找到染色体的方向，找到中心，测量臂长、灰度等参数，然后加以识别。染色体分类系统的框图如图10-35所示。目前有资料谈到染色体的正确识别率为93.7%。

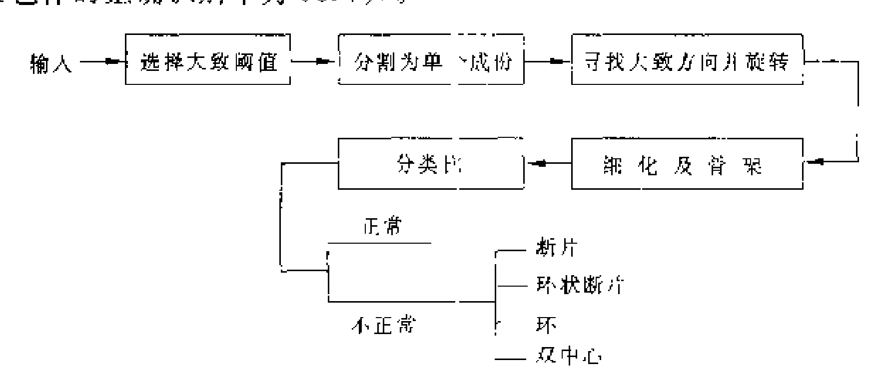


图10-35 染色体分类系统框图

除染色体分类，在医学中的应用还有血球分类。目前有的医院使用5类分类器，可以做到95%的正确分类。分类方法与染色体分类大致相同。此外，还有细胞分类、X射线透视照片分析等等。

10.5.3 模式识别在自动检测中的应用

模式识别的一个较为广泛的应用领域是自动检测，它包括自动视觉检查及工业零件的自动识别等等。在生产过程中，为了排除次品和查出与故障有关的零件隐患等，要在生产过程中用目测方法对产品进行外观检查，但目测法有因人而异和漏检的问题。为此，有必要研制能在一定的标准条件下定量的一个不漏地进行检查的装置，这就是利用模式识别技术的自动外观

检查装置。

目前已实用的自动外观检查装置有很多，如漆包线自动外观检查装置，彩色显像管阴罩的自动外观检查装置，二极管基片检查、电话交换机继电器接点检查以及印刷电路板自动外观检查等等都已付诸实用。下面简略介绍其中几种检测原理。

1. 印刷电路板自动外观检查装置

在电子技术高度发展的今天，几乎所有的电气装置中都要用到印刷电路板，特别是电子计算机中的印刷电路板，已达到高密度多层化。因此，用目测法检查缺陷相当困难甚至已不可能。大部分印刷电路板的缺陷都是由一些细微的疵点造成的，只要从图案中检查出细微的图案差错，就能防止大部分缺陷。在识别微细伤痕时，要从可能含有伤痕的输入图像中先做出不包含任何伤痕的图像作为模板，这种模板称为准正常图像。然后将准正常图像与输入图像进行比较，识别出与准正常图像有差别的部分，判定其为疵点或伤痕。

准正常图像的制作方法如图 10-36 所示。图(a)是输入图像，其中有微小的斑疵。为消除图像中的疵点，首先把图像放大，如图(b)、(c)把黑色部分放大，紧接着再把黑色部分等量缩小，则图中的白色疵点去掉了。然后均匀放大白色部分，再等量缩小，如图(d)，(e)所示，此时黑色疵斑也去掉了。最后得到图(e)所示的准正常图像。

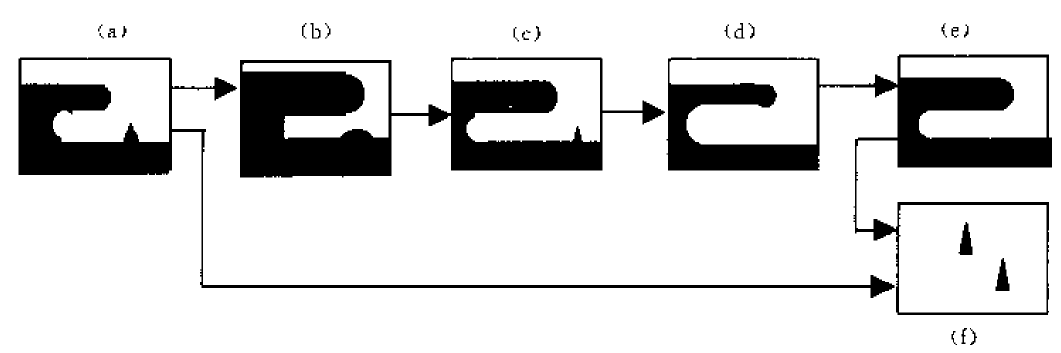


图 10-36 印刷电路板缺陷检查原理

印刷电路板外观检查装置的方框图如图 10-37 所示。这种装置用摄像机来检查，首先把图像转换成 240×320 个点阵像素，然后作二值处理，提取细微部分，从而判定伤痕。这种方法处理一张图像约需$\frac{1}{60}$s。

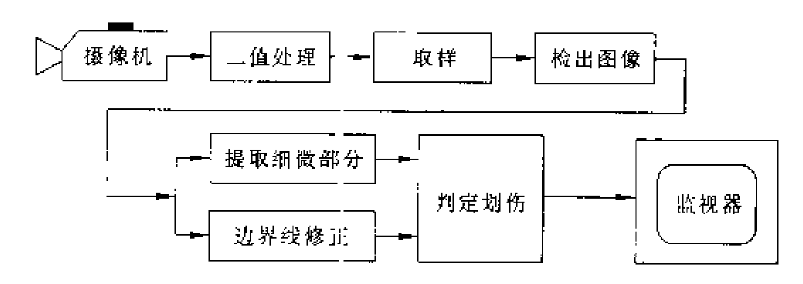


图 10-37 印刷电路板自动外观检查装置框图

2. 彩色显像管阴罩底板的自动外观检查

彩色显像管阴罩的作用是要确保每种颜色的所有荧光点仅受其相应的电子束的轰击，所以阴罩的制造精度直接关系到显像管色彩的准确度与图像的鲜明度。阴罩板上精确地开有几十万个宽 0.3mm，长 0.6mm 的带式图案。

在阴罩的制作过程中，采用光刻法。首先在钢板上涂上光致抗蚀剂，将要使用的玻璃底板

紧贴于阴罩的原板上曝光，以便使玻璃底板上的图案转印到阴罩原板上，所以要求玻璃底板图案尺寸精度高，而且无缺陷。

图 10-38 示出了玻璃底板和带式图案的缺陷。图(a)的玻璃底板上有两张底板图案。带式图案中的缺陷示于图(b)。由于 10μm 以上的缺损，边缘突起，内部缺陷，多余部分以及带式图案的尺寸误差等均与阴罩质量的低劣有关，所以要对上述各项进行全面检查，然后予以修正。过去，上述检查都是在显微镜下用目测法进行，现在可用图像识别技术自动检出缺陷。

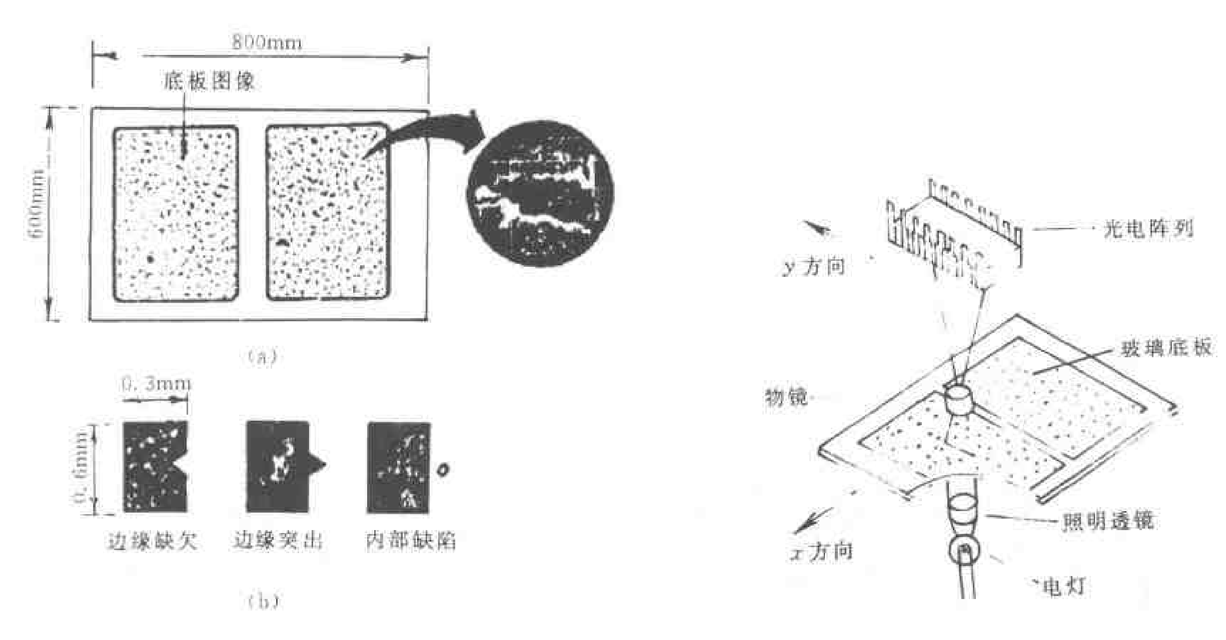


图 10-38 玻璃底板及带式图案缺陷示例

图 10-39 底板图像缺陷检查原理

检测原理如图 10-39 所示。从玻璃底板背面照明，用物镜放大底板图案，然后用光敏阵列器件检测。该阵列器件是在纵向排列有 128 个大小为 25μm×25μm 的光敏元件。玻璃底板在 x 方向上往复运动，这时，电光源、照明透镜、物镜及光电阵列器件组成一个整体，在 y 方向上间歇地以一定节距作进给运动，这样就可全面检测底板图案。检查内容如下：

1)测量图案宽度、长度及中心线的偏差量；

2)提取超过 10μm 的微小部分；

3)检查图案的全部缺陷。

为了提高速度用了 5 个检测头，每个检测头以 300 个图案/秒的速度对带式图案进行检查。检测信息送入计算机，用模板匹配法判断是否合格。

3. 漆包线自动外观检查

漆包线大量用于电力与电子工业中。若在漆包线表面上有漆疙瘩或有掉漆缺陷，漆包线的绝缘性能就会下降或绕线不齐，这会降低产品性能。自动检测装置可检测的漆包线上最小缺陷尺寸为 0.1mm×0.03mm。它的基本检测原理是用光均匀地向线上一点照明，如果有缺陷则会产生光的漫反射，没有缺陷则不会产生漫反射，由此可通过检测光的漫反射识别出缺陷。

4. 电话交换机转换开关或继电器接点自动外观检查

电话交换机转换开关是以镀有 Au-Pd 的小铜片作为接点的，为了减少接触电阻用喷沙法把表面毛刺打光。接点大小为 0.35mm×0.8mm。这种接点表面会产生如图 10-40 所示的各种缺陷。由于位置偏差、尺寸不对、表面伤痕、附着杂质、形状不规则等等有可能导致接点接触不良。所以要进行严格检查，排除次品。焊接溶渣对焊点本身并无不良影响，但长期使用会脱落，

附着在接点接触部分或其他机构上。为此,也要剔除有焊接溶渣的接点。

在检查中利用显微镜,用照明灯从正上方对焊接点表面照明并由摄像机摄取图像。此时,平坦部分明亮,点轮廓、伤痕、杂质、焊接溶渣等缺陷部分会使照明光的一部分漫反射,从而使向正上方反射回去的那部分光的分量减少。把电视摄像机检出的接点的轮廓像在 x 方向予以投影,作出暗区频率分布曲线,便可做精密检查。基于这一点,设制内框和外框,并规定其中有 50μm 以上的暗区就定为缺陷。如图 10-41 所示,从中也可检查出内框有伤痕,外框有溶渣。这种装置 32 个接点的检查时间为 6 秒。

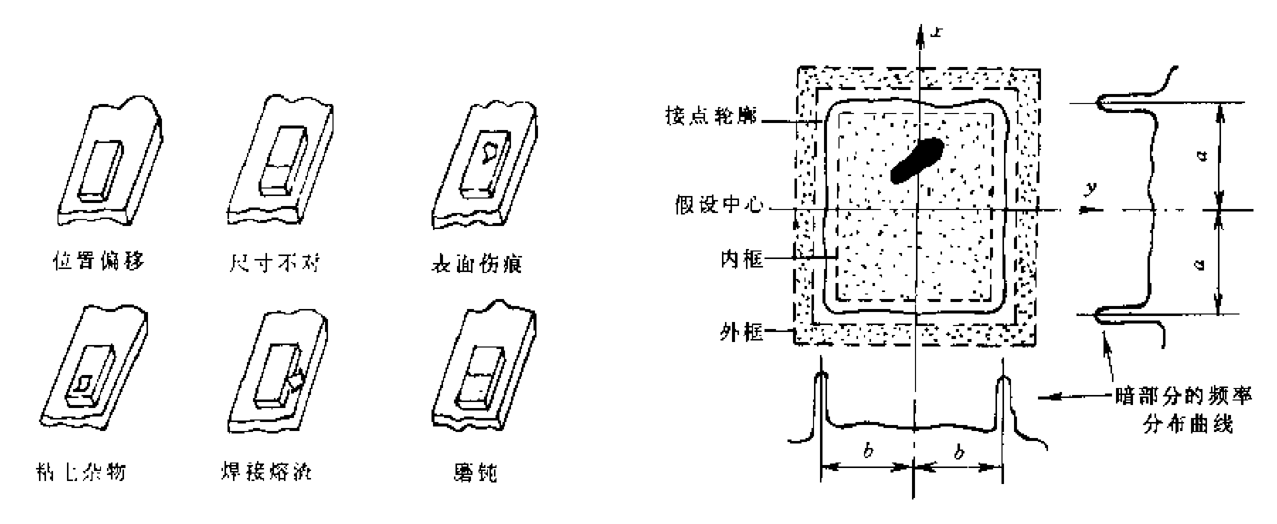


图 10-40　接点的各种缺陷示例　　　图 10-41　转换接点缺陷检出原理

除上述几种检查外,模式识别还用于二极管芯片的自动外观检查。它可自动筛选出有潜在缺陷的芯片。这个检查装置可检出 0.1mm 以上的缺陷,其检查速度为每秒 10 个芯片。

模式识别在工业及其他领域中的应用还有焊缝缺陷自动识别、机械零件自动分类、汽车类型识别、信件自动分拣、晶体管组装系统中基片位置自动识别等等大量的用途。在遥感中对航片及卫片的分析、分类也是模式识别的一个重要领域。同时模式识别也是研制带有视觉及听觉系统的机器人的不可缺少的组成部分。尤其是当今人机自动交互的研究中,模式识别几乎是最主要的技术。随着模式识别理论的日益深入的研究,它的应用领域也会日益扩大,它将成为人类生产实践中不可缺少的有力工具。

思　考　题

1. 试说明统计模式识别的原理。

2. 线性判别函数在两类分类器中如何应用?

3. 在什么情况下可用贝叶斯分类法?

4. 边缘抽取中的断线如何处理?举例说明膨胀与收缩算法的操作与作用,试编一小程序处理之。

5. 试述句法模式识别的原理,并画出其原理框图。

6. 考虑两类决策问题,如果 $X>Q$,判决为 W_1,否则,判决为 W_2,指出错误概率:

$$P = P(W_1)\int_{-\infty}^{Q} P(X|W_1)\mathrm{d}X + P(W_2)\int_{Q}^{\infty} P(X|W_2)\mathrm{d}X$$

对上式求导，指出 P 极小的条件是 Q 满足：

$$P(Q|W_1)P(W_1) = P(Q|W_2)P(W_2)$$

7. 如何用最小风险求类别？
8. 试设计 -个三类贝叶斯分类器，画出其框图，并说明其操作。

附录 10 图像采集卡的参数及使用(供光盘中软件参考)

为使读者充分理解和掌握本书的内容，本书根据笔者多年的教学和科研的体会特编写了一些实用的程序，并提供一套完整的数字图像处理的软件。该软件可以用于教学演示和实验，也可以供自学者参照理解全书的内容及为初学者提供编程练习指导。本软件图像采集卡选用中国科学院自动化研究所科技佳公司的 CA-CPE-1000。现对该卡的技术特点及使用方法介绍如下。

1. 技术特点及指标

CA-CPE-1000 是基于微型计算机 PCI 总线结构的彩色(黑白)图像采集卡．它采用先进的数字解码方式，将标准输入的 PAL 制、NTSC 制式、SECAM 制的复合彩色(或黑白)视频信号或 S-Video 信号(即 Y-C 分离信号)数字化，经解码后转换为适于图像处理的 RGB-24bit 格式的数字信息，然后通过 PCI 总线实时传送到 PC 机系统内存(或视频显示缓冲区)。其技术参数如下：

· 标准 PAL、NTSC 或 SECAM 制彩色/黑白输入信号；
· 六路 CVBS 输入软件选择或三路 Y/C 输入选择；
· 支持 RGB24、RGBI6、YUV16 格式，及 8bit 黑白格式；
· 亮度、色调、饱和度、对比度软件可调；
· 采集图像大小、位置、压缩比软件设定；
· 图像采集显示分辨率最大 768×576×32bit；
· 采集图像由 VGA 卡显示，实现图像和菜单同屏显示；
· 灰度精度±1/256，点阵抖动(pixel jitter)不大于 10ns；
· 采集方式灵活(按单场、单帧、连续帧及间隔几帧等方式采集)；
· 一路辅监输出；
· 稳定接受录像机视频信号；
· 提供 DOS、Windows 3.x、Windows 95 环境下开发库及 Video for Windows 驱动。

该图像卡结构框图如图 1 所示。一个完整的实时图像采集处理系统，如图 2 所示。

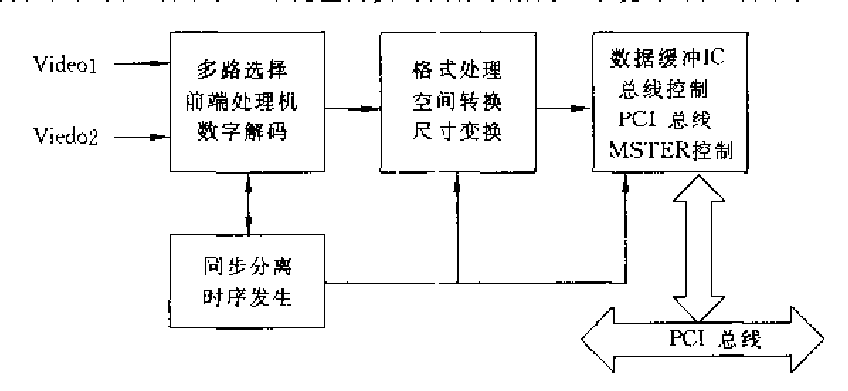


图 1 CA-CPE-1000 图像采集卡结构框图

CA-CPE-1000 图像采集卡采用总线控制技术，图像传送速度高达 60MB/s，可实现摄像机图像到计算机内存实时传送，连续相邻帧的图像精确到场。由于采用了高精度 Gen Lock 技术和线性箝位技术，采集的图像

图 2　图像采集处理系统示意图

点阵位置精度高，A/D 转换后的数字视频信号误差小，适用于各种高精度工业和科学图像处理、多媒体的压缩等研究开发领域。

图像采集点阵为矩形，大小和位置可选，从 4×4 到 768×576，X、Y 方向分别由软件 4 为步长选择，实现 AOI(Area of Interesting)。

由于 CA-CPE-1000 将图像直接传送到主机的内存，可连续存储相邻的多帧图像。同时，图像处理算法在主机内存执行，有利于提高处理速度，充分发挥越来越高的 CPU 潜力，并便于用户编程。该卡用 VGA 卡可同时显示图像和菜单，图像显示速度取决于 VGA 卡数据传输速度。此外，该卡采用中断方式将图像数据传送和用户程序灵活地连接起来。CA-CPE-1000 也可设置成黑白图像采集方式。

2. 系统配置要求

(1) 硬件配置

主机板：带 PCI 插槽，支持新的 PCI 规范；

CPU：586 系列；

注意：某些 486 的机器上能且仅能在 DOS 环境下一次采集一帧，不能连续采集。

内存：在 DOS 环境下使用内存为 4MB 以上；

在 Windows 3.x 环境下使用内存为 8MB 以上；

在 WIN95 环境下使用内存为 16MB 以上。

显示卡：显示卡的速度主要影响实时显示的性能。若显示卡速度不够，实时显示时会出现图像采集不全或死机现象。建议选用 S3 系列的 VGA 卡，例如 S3、Trio64V+、S3WINFAST 显示卡。在 WIN95 下使用 S3 卡提供的 S3.INF 构造显示驱动程序。

显示卡帧存：采集真彩色图像要求 2M 以上；采集灰度图像要求 1M 以上。

(2) 软件环境配置

① CA-CPE-1000 是将图像直接采集到计算机的扩展内存，因此对图像卡进行操作时必须和系统扩展内存打交道。在使用 CA-CPE-1000 时，务必在 CONFIG.SYS 中加入扩展内存管理设备驱动程序，如 MS-DOS 提供的 HIMEM.SYS。

例：DEVICE=C:\DOS\HIMEM.SYS

② CA-CPE-1000 采用内存映像方式访问硬件，因此当在 DOS 环境下使用 CA-CPE-1000 时，最好不要使用 EMM386.EXE。若一定需要在 CONFIG.SYS 中保留 EMM386.EXE，可采用如下方式：

DEVICE=C:\DOS\EMM386.EXE/X=D000~D3FF

注意：在 Windows 环境下使用没有此限制。

③ 由于 SMARTDRV.EXE 将占用 2MB 内存，而 CA-CPE-1000 也至少应分配 2MB 内存，同时需要为 Windows 3.x 保留 4MB 内存，因此，若在 Windows 3.x 下使用 CA-CPE-1000 彩色图像卡采集系统，且主机内存小于等于 8MB 时，不要使用 SMARTDRV.EXE。

④ 关于 Windows 3.x 环境下实时显示

采集卡在 Windows 3.x 环境下，通过 DCI 接口实现实时显示。

DCI 接口是一种设备驱动程序规范，该接口给系统服务提供了一种直接写视频硬件的方法，DCI 功能均由显示卡生产厂家提供。在 DCI 用户与显示卡驱动程序之间有一个标准 DCI 管理程序，称为 DCIMAN.DLL。若显示卡支持 DCI 接口，则在显示卡驱动程序的安装过程中会提供 DCIMAN.DLL，并拷贝在 Windows 的 SYSTEM 目录下。

注意：当 Windows 3.x 处于 16 色 VGA 显示模式下时，所有的显示卡不具备 DCI 功能。

3. 安装

（1）硬件安装

① 关闭计算机电源，拔掉所有电源线；关闭视频输入设备电源。

② 打开计算机机箱。

③ 将 CA-CPE-1000 图像采集卡插入任何一个空闲的 PCI 插槽。请注意，该 PCI 插槽必须支持主控方式，相关信息请查阅计算机用户手册。

④ 关闭计算机机箱。

⑤ 拿出随卡附送的视频输入线，D 型头一端与 CA-CPE-1000 的 15 孔插座相连，BNC 插头一端与视频输入设备相连（黄色标记线为缺省输入线）。

⑥ 打开视频输入设备电源开关。

⑦ 打开计算机电源开关。

本图像采集卡可以接受 CVBS 视频信号和 S-Video 视频信号（Y/C 分离信号），最多可连接 6 路 CVBS 视频信号或 3 路 S-Video 视频信号。连接方式分别如图 3 和图 4 所示。

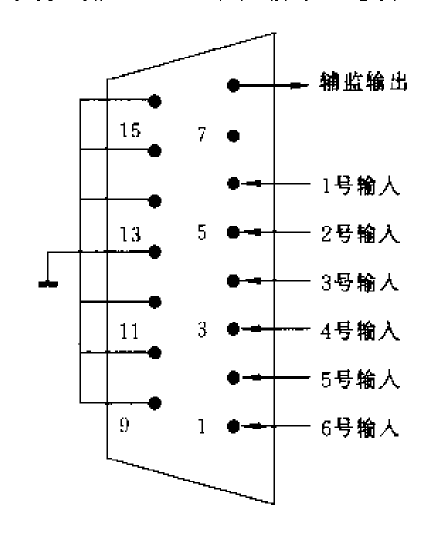


图 3　6 路 CVBS 信号输入连接

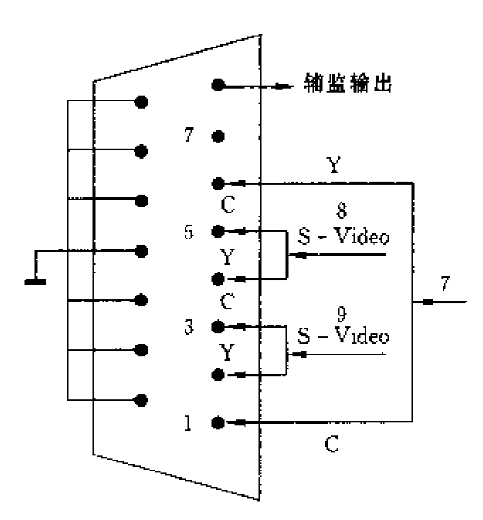


图 4　3 路 S-Video 信号输入连接

（2）软件安装

CA CPE-1000 随卡提供的软盘上有两个文件：

CPE1000.EXE 自解压缩文件

README.TXT 对 CPE1000.EXE 使用方法的简单说明

① 在硬盘上建立一个目录，如 MDC:\CPE1000；

② 将随卡软盘文件拷贝到硬盘目录下，如 COPY A:*.* C:\CPE1000

③ 执行硬盘上的文件 CPE1000.EXE，程序自解压缩，并创建目录，拷贝文件。安装结束，CA-CPE-1000 系统软件即存在于当前目录下。

④ 若需要在 Windows 环境下使用采集卡，请将 VxD.386(\WINSYS 目录下)拷贝到 Windows 3.x 的 SYSTEM 目录下。打开 Windows 3.x 的配置文件 SYSTEM.INI，找到[386Enh]节，加入一条语句：device = vxd.386；在 SYSTEM.INI 中加入新的节，名为[CA-IMAGE]；在[CA-IMAGE]节中加入一条语句：FrameSize

800h。

5. 若需要在 Windows 95 环境下使用采集卡，请将 VxD. vXd(\CPEW. 95 目录下)拷贝到 Windows 95 的 SYSTEM 目录下。打开 Windows 95 的配置文件 SYSTEM. INI，找到[386Enh]节，加入一条语句：device=vxd. vxd；在 SYSTEM. INI 中加入新的节，名为[CA IMAGE]；在[CA-IMAGE]节中加入一条语句：FrameSize=800h。

本系统软件包括：DOS 环境(\CPE 目录下)、Windows 3. x 环境(\CPEW 目录和\WINSYS 目录下)及 Windows 95 环境下的系统软件。

\CPE 目录下的 DOS 环境系统软件包括：

\BCLIB 子目录下的程序是对 DOS 环境库函数使用的简单示例，并可测试采集卡基本功能，如实时显示及实时采集到内存并显示(至少 9 个文件)。

TCPE. C	测试库函数功能源文件
TCPE. EXE	测试库函数执行文件
USERCPE. H	库函数输出头文件
CPE_LIB. LIB	CA-CPE-1000 采集卡函数库
TCPE. PRJ	测试库函数功能工程文件
MCPE. C	采集灰度图像的测试源文件
CPE_MONO. PRJ	采集灰度图像的工程文件
CPE_MONO. EXE	采集灰度图像的测试执行程序

\BIN 子目录下的程序是 DOS 环境下演示程序(至少 9 个文件)。

CHLIB	演示程序菜单字库
CPE_MENU. C	演示菜单源程序
MAIN. C	演示程序测试库函数源程序
DEMO. EXE	演示程序测试库函数执行文件
MENU. H	演示程序头文件
USERCPE. H	库函数输出头文件
CPE_LIB. LIB	CA-CPE-1000 采集卡函数库
DEMO. PRJ	演示程序测试库函数工程文件

\CPEW 目录下的 Windows 3. x 环境系统软件包括(至少 10 个文件)：

TCPEW. C	Windows 3. x 下库函数演示源文件
PAL. C	辅助源程序
TCPEW. DEF	定义文件
CPEW. DLL. DLL	Windows 3. x 下动态连接库
TCPEW. EXE	Windows 3. x 下测试库函数可执行文件
TCPEW. H	头文件
USERCPEW. H	Windows 3. x 下动态连接库输出头文件
TCPEW. PRJ	Windows 3. x 下测试库函数工程文件
TCPEW. RC	资源文件

\CPEW. 95 目录下 Windows 95 环境 32 位系统软件包括(至少 12 个文件)：

vxd. vxd	Windows 95 下使用的内存管理虚拟驱动程序
CPEW16. DLL	运行时所需的动态连接库
CPEW32. DLL	
CPEW32. LIB	Visual C—— 4. x 下所使用的入口库
CPEW32B. LIB	Borland C++ 5. 0 下所使用的入口库
CPEW32. H	为进行 Win32 编程所需的 C 头文件

TCPEW32. C　　演示程序的 C 源代码

TCPEW32. H　　头文件

TCPEW32. RC　　资源文件

TCPEW32. EXE　　演示程序可执行文件

README. TXT　　最新版本更新说明

\WINSYS 目录下文件：

VXD. 386　　Windows 3. x 下内存管理虚拟驱动程序

README. TXT　　VXD. 386 使用方法简单说明

\DOC 目录下文件

CPE1000. DOC　　本手册(Word6. 0 版本)

(3) 库函数的使用与编译

① DOS 环境

工程文件 TCPE. PRJ 或 CPE_MONO. PRJ，工程文件中包括如下文件：

源程序 TCPE. C 或 CPE_MONO. C

CA-CPE-1000 函数库,CPE_LIB. LIB

C 语言程序中包含头文件 USERCPE. H

DOS 下函数库由 Borland C 3. 1 编写，因此用户程序必须在相同编译环境下编译连接。此外，编译采用大模式，连接选项应设置为忽略大小写。编译时确保 CPE_LIB. LIB 存在于 BC 可访问的目录中。

② Windows 环境

工程文件 TCPEW. PRJ，该文件属性为 Windows App，工程文件中包括：

演示源程序 TCPEW. C

辅助源程序 PAL. C

资源文件 TCPEW. RC

定义文件 TCPEW. DEF

Windows 环境下提供. DLL 库，可在 Windows 的任何编译器中使用。编译采用大模式，连接选项应设置为忽略大小写。

③ Windows 95 环境

需创建 TCPEW32 工程文件，该工程文件属性为 Win32App，不使用 C++及类库扩充。工程文件中包括如下文件：

演示源程序 TCPEW32. C

资源文件 TCPEW32. RC

输入库文件 CPEW32. LIB 或 CPEW32B. LIB

若您使用 Visual C++ 4. x 或更高版本，请将目标代码与 CPEW32. LIB 连接；若使用 Borland C++ 5. 0 或更高版本，请将目标代码与 CPEW32B. LIB 连接。

执行 TCPEW32. EXE 时请注意确认 CPEW32. DLL 与 CPEW16. DLL 均在可访问的目录之内。

4. DOS 环境下软件说明

(1) 软件概述

系统在 DOS 下提供了集成环境程序 DEMO. EXE(\CPE\BIN 目录下)，该环境演示了系统的采集及文件管理等功能。如仅需采集图像生成文件，可直接使用集成环境。

除上述程序之外，系统还提供了一个 DOS 环境下的基本功能演示程序 TCPE. EXE(\CPE\BCLIB 目录下)。该程序没有菜单，仅顺序完成实时显示、实时采集、文件保存及回放功能。该程序功能简单，源程序易读，对本采集卡库函数不熟悉的用户可从 C 语言源程序 TCP E. C 入手，做为例子程序阅读。

在此要说明的是：CA-CPE-1000 图像卡系统软件中，规定了两种图像文件：单帧图像文件和序列图像

文件。

彩色单帧图像文件扩展名为.RAW,文件头部结构如下:

起始位置	长度(Byte)	注　释
00	2	图像宽度
02	2	图像宽度
04	1	蓝
05	1	绿
06	1	红
07	1	α(目前无用)
08	$(n-1)\times 4$	$(n-1)$个像素依次按由左至右,由上至下的顺序排列

黑白单帧图像文件扩展名也为.RAW,但其图像的宽度和高度在文件末尾,结构如下表所示:

起始位置	长度(Byte)	注　释
00	1	一个像素
01	$(n-1)\times 1$	$(n-1)$个像素依次按由左至右,由上至下的顺序排列
n	2	图像宽度
$n+2$	2	图像宽度

序列图像文件扩展名为.SRS、文件内容是一系列单帧图像文件的文件名,真正的图像还是存储在单帧图像文件中,文件结构如下:

起始位置	长度(Byte)	注　释
00	2	图像宽度
02	2	图像宽度
04	n	一系列单帧图像文件名

在存储序列图像文件时,图像信息按帧存储在一系列单帧图像文件中,每个单帧图像的文件名由序列图像文件名后加上 001、002 等连续的数字组成,这要求序列图像文件名应少于 5 个字母。在调出序列图像文件时,根据序列图像文件中存储的单帧图像文件名,依次打开这些文件,并读出文件中的图像信息按帧放到扩展内存中。

注意:序列图像文件名应少于等于 5 个字母,如 xxxxx.SRS。

(2) 集成环境使用说明

运行\CPE\BIN 目录下的 DEMO.EXE 程序,进入 DOS 环境下的集成环境。集成环境演示了包括采集、文件处理、选择信号源、设置窗口等大部分库函数功能。在集成环境中,高亮的菜单项表示是当前选中的菜单项,在高亮的菜单项处按 ENTER(回车)键则进入该项菜单对应的功能。按"←"键和"→"键可以在菜单中左右移动。

· Real time grab 实时显示,并调节采集参数.按 ENTER 键开始采集。用户通过按 F1、F2 调节对比度;F3、F4 调节亮度;F5、F6 调节饱和度;F7、F8 调节色调;按 ESC 键则退出实时采集状态,返回主菜单。

· Save grab date 实时显示并以.BMP 文件格式存至 C:\。选中该菜单项并按 ENTER 键后,首先进入实时显示状态。当用户选定采集视野,按 ESC 键退出实时显示状态同时采集一帧图像到内存,将采集到的图像数据以.BMP 格式存至 C 目录,名称为 1.BMP,文件存盘完毕后,返回主菜单。

注意:若存盘文件较大,则存盘需要消耗一定的时间,请耐心等候,在此期间不要随意按键。可以按一下"←"键或"→"键,这样当存盘完毕后,主菜单会有所变化,据此可以得知存盘操作已经完毕。

· Display capture data 实时采集图像至内存并显示所采集到的数据。选中该菜单项并按 ENTER 键后，则采集一帧图像至扩展内存，同时显示该帧图像数据，显示完毕后即返回主菜单。

Write line to XMS 对内存的数据按行读写。选中该菜单项并按 ENTER 键后，则采集一帧图像至扩展内存，对该帧图像按行进行读写操作，之后显示经过这样修改的该帧图像数据，显示完毕后即返回主菜单。

· Set grab window 设置采集窗口。选中该菜单项并按 ENTER 键则激活此功能项。在这种情况下，按 G 键缩小采集窗口，按 H 键放大采集窗口。若确认采集视野，按 Q 键则退出此功能项，返回主菜单。

· Read and write window 对内存的数据进行读写窗口操作。选中该菜单项并按 ENTER 键后，则采集一帧图像至扩展内存，对该帧图像按窗口(块)进行读写操作，之后显示经过这样修改的该帧图像数据，显示完毕后即返回主菜单。

· Capture serial frame 实时采集序列图像至内存并在 C 盘根目录存成序列图像文件格式。选中该菜单按 ENTER 键则开始进入实时显示状态，确定采集视野后按 ESC 键则采集 3 帧序列图像至扩展内存。将这三帧以序列文件的形式存储到 C 盘根目录。其具体文件包括：1.SRS、1001.RAW、1002.RAW 及 1003.RAW。存盘完毕后按任意键，显示刚刚采集的第一帧图像，再按任意键显示下一帧，直至显示完采集的三帧图像，返回主菜单。

· 用 Save as rawfile 实时采集图像至内存并以.RAW 格式存放在 C 盘根目录下。选中该菜单按 ENTER 键则进入实时显示状态，确定采集视野后，按 ESC 键则将采集一帧图像至扩展内存，并将内存中的图像数据以.RAW 文件格式存储到 C 盘根目录。文件名称为 1.RAW。存盘完毕即返回主菜单。

· CVBS source select 切换信号源。当该菜单高亮时用户通过键入数字可以切换信号源。

· Gain control analog 调节增益控制。选中该菜单项并按 ENTER 键则激活此功能项。按 G 键降低增益控制，按 H 键提高增益控制，按 Q 键退出此功能项，返回主菜单。

Clamplevel Control 调节箝位电平。选中该菜单项并按 ENTER 键则激活此功能项。按 G 键降低箝位电平，按 H 键提高箝位电平，按 Q 键退出此功能项，返回主菜单。

· IncrementDelay_7110 调节系统延迟参数。选中该菜单项并按 ENTER 键则激活此功能项。按 G 键降低系统延迟参数，按 H 键提高系统延迟参数，按 Q 键退出此功能项，返回主菜单。

· ClampBegin Control_7110 调节箝位电平。选中该菜单项并按 ENTER 键则激活此功能项。按 G 键降低箝位电平上跳沿时钟，按 H 键滞后箝位电平上跳沿时钟，按 Q 键退出此功能项，返回主菜单。

· ClampStop Control 调节箝位电平下跳沿时钟。选中该菜单项并 ENTER 键则激活此功能项。按 G 键提前箝位电平下跳沿时钟，按 H 键滞后箝位电平下跳沿时钟，按 Q 键则退出此功能项，返回主菜单。

· Return to DOS 选中该菜单按 ENTER 键，退出集成演示环境，返回 DOS 操作系统。在主菜单的任何位置按 ESC 键，则菜单总是自动跳转到这一菜单项。在该菜单位置按 ESC 键，退出集成演示环境，返回 DOS 操作系统。

(3) 函数库详解

CA-CPE-1000 函数库包括四类函数，分别为：系统控制函数、文件管理函数、VGA 函数和数据传输函数。

① 系统控制函数

A 类. 初始化函数

(1.1) 初始化 CA-CPE-1000 图像采集卡

```
int Init_CPE-1000 (unsigned int wReserveSize)
```

(1.2) 关闭 CA-PE-1000 图像采集卡

```
void Close_CPE1000 (char outinformation[80],\
                    int retmode)
```

(1.3) 获取 CA-CPE-1000 所在 PCI 插槽的中断号

```
int Get_CPE1000_IRQnum (void)
```

B 类. 设置函数

(1.4) 设置采集图像的视频位置及大小

int Set_Grab_Window (int xl,int y1,int x2,int y2)

(1.5) 设置采集图像的目标大小

int Set_Target Window (int wDestW,int wDestH)

(1.6) 设置"实时显示"时功能键号

void Function_KEY (int conUp,int conDn,int briUp,\
  int birDn,int satUp,int satDn,int hueUp,int hueDn,\
  int breakKey)

(1.7) 选择输入信号源

void Set_Input_source (int vin_select)

(1.8) 设置采集方式,真彩色或灰度图像

int Set_Pixel_Format (int p_format)

(1.9) 设置采集对比度、亮度、饱和度及色调

void Set_Contrast (unsigned char)

void Set_Brightness (unsigned char)

void Set_saturation (unsigned char)

void Set_Hue (unsigned char)

(1.10) 获取当前对比度、亮度、饱和度及色调

unsigned char Get_Contrast (void)

unsigned char Get_Brightness (void)

unsigned char Get_saturation (void)

unsigned char Get_Hue (void)

C 类. 采集函数

(1.11) 实时采集图像至显示卡帧存体,即实时显示

int CPE_Grab_VGA (int dispX,int dispY)

(1.12) 实时采集图像至扩展内存

int CPE_Grab_Xms (int sFrames,int totalFrames)

② 文件管理函数

(2.1) 读单帧图像文件至扩展内存

int ReadRaw2XMS (char * fname,int nframe)

(2.2) 写单帧图像.RAW 格式文件

int WriteXMS2Raw (char * fname,int nframe)

(2.3) 读序列图像文件至扩展内存

int ReadSrs2XMS (char * fname,int frameNum)

(2.4) 写序列图像文件

int WriteXMS2BSrs (char * fname,int sframe,int eframe)

(2.5) 写.BMP 格式文件

int WriteXMS2Bmp (char * szBmpfile,int nframe)

③ VGA 函数

(3.1) 初始化 VGA

int InitVesa (int wVesaMode)

(3.2) 关闭 VGA

int CloseVesa (void)

(3.3) 显示图像

int DispXMS (unsigned int wMode,unsigned int wFrame,\

int start_X,int start_Y)

(3.4) 640×480 分辨率下显示灰度图像

void DISP_640×480 (int x,int y,int nframe,\
int Frame_Field_Mark)

(3.5) 800×600 分辨率下显示灰度图像

void DISP_800×600 (int x,int y,int nframe,\
int Frame_Field_Mark)

④ 数据传输函数

(4.1) 读行

int ReadHLine (int nframe,int y,int x1,int x2,\
unsigned char far * buff)

(4.2) 写行

int WriteHLine (int nframe,int y,int x1,int x2,\
unsigned char far * buff)

(4.3) 读窗口

int ReadWindow (int nframe,int x1,int y1,int x2,int y2,\
unsigned char far * buff)

(4.4) 写窗口

int WriteWindow (int nframe,int x1,int y1,int x2,int y2,\
unsigned char far * buff)

(4.5) 读像素

unsigned long ReadPixel (int nframe,int x,int y)

(4.6) 写像素

int WritePixel (int nframe,int x,int y,unsigned long val)

下面详细解释各个函数：

① 系统控制函数

(1.1) int Init_CPE1000 (unsigned int wReserveSize)

功能：初始化 CA-CPE-1000 图像采集卡，必须在执行其他函数之前调用。

参数：wReserveSize：用户要保留的扩展内存大小，以 kB 为单位。

返回： 0　　成功；

-1　　扩展内存管理设备驱动程序没有安装，
请您在 CONFIG.SYS 文件中加入 HIMEM.SYS；

-2　　扩展内存不够；

其他(<0) 扩展内存管理错误。

举例：int result；

result = Inut_CPE1000 (2048)。

说明：调用该函数后，对 CA-CPE-1000 及扩展内存进行初始化。将为用户保留一定的扩展内存，保留的内存大小由用户决定。

(1.2) void Close_CPE1000 (char outinformation[80],\
int retmode)

功能：关闭 CA-CPE-1000 图像采集卡。

参数：outinformation：关闭采集卡的有关信息；

retmode：在屏幕上显示 outinformation 信息，设置为 0 时，exit(0)退出；设置为其他值时，退回到用户应用程序。

返回：无。

举例：Close CPE1000("Every thing is OK!!!",0)。

说明：该函数释放由 Init_CPE1000()占用的扩展内存。

(1.3) int Get CPE1000 IRQnum(void)

功能：获取 CA-CPE-1000 所在 PCI 插槽占用的中断号。

参数：无。

返回：CA-CPE-1000 所在 PCI 插槽的中断号。

举例：int IRQnum;

IRQnum Get CPE1000 IRQnum()。

(1.4) int Set_Grab_window(int xl,int y1,int x2,int y2)

功能：设置采集图像在视野中的偏移位置与大小，即采集窗口。

参数：(x1,y1)：采集图像的左上角坐标；

(x2,y2)：采集图像的右下角坐标。

实际采集窗口大小为

宽＝x2－xl＋1；高＝y2－y1＋1。

返回：－1 长度或宽度为零；

－2 长、宽大于屏幕长、宽；

＞0 最大可采集帧数。

举例：int Total_Frames;

Total_Frames＝Set_Grab_Window(0,0,639,479)。

(1.5) int Set_Target_Window(int wDestW,int wDestH)

功能：设置采集图像目标窗口大小。

参数：wDestW：目标窗口宽度；

wDestH：标窗口高度。

返回：－1 目标窗口大于采集窗口；

－2 目标窗口高度、宽度未以 4 为倍数；

＞0 最大可采集帧数。

说明：由函数(1.4)设置采集窗口采集摄像头视野的大小。

本函数设置最终得到的图像大小。目标窗口只能小于等于采集窗口，当小于采集窗口时，即为缩小采集。如等于采集窗，或不使用本函数，即按采集窗口原大采集。

(1.6) void Function_KEY(int conUp,int conDn,int briUp,\
int birDn,int satUp,int satDn,int hueUp,int hueDn,\
int breakKey)

功能：设置实时显示时调节采集参数所用的键。

参数：conUp：增加对比度的值所用的键号；

conDn：减小对比度的值所用的键号；

briUp：增加亮度的值所用的键号；

briDn：减小亮度的值所用的键号；

satUp：增加饱和度的值所用的键号；

satDn：减小饱和度的值所用的键号；

hueUp：增加色调的值所用的键号；

hueDn：减小色调的值所用的键号；

breakKey：退出实时显示所用的键号。

返回：无。

(1.7) void Set_Input_source(int vin_select)

功能：选择输入信号源。

参数：vin_select：信号源选择，取值范围：0～8.

返回：无。

举例：Set_Input_Source(0)；

说明：取值信号源

0～5 CVBS　No. 1～ 6 输入

6　S-Video　Y：No. 1　C：No. 6

7　S-Video　Y：No. 3　C：No. 2

8　S-Video　Y：No. 5　C：No. 4

其中，No. 1～6 为输入信号线的编号。

(1.8) int Set_Pixel_Format(int p_format)

功能：设置采集方式，真彩色或灰度图像。

参数：p_format：采集方式，

0X00　真彩色采集方式；

OXC3　256 级灰度采集方式。

返回：返回值为在该设置情况下，最大可采集帧数。

>0 成功，最大可采集帧数。

举例：int Total_Frames；

Total_Frames =Set_Pixel_Format(0x00)。

说明：系统初始化时，默认设置为真彩色采集方式。

注意：真彩色采集方式下一个像素由 32bit 体现，灰度采集方式下一个像素由 8bit 体现。

(1.9) void　Set_Contrast(unsigned char contrast)

void　Set_Brightness(unsigned char brightness)

void　Set_saturation(unsigned char saturation)

void　Set_Hue(unsigned char hue)

功能：设置采集对比度、亮度、饱和度及色调，

参数：对比度、亮度、饱和度及色调对应的设置值

有效范围：0～255。

返回：无。

(1.10) unsigned char Get_Contrast(void)

unsigned char Get_Brightness(void)

unsigned char Get_Saturation(void)

unsigned char Get_Hue(void)

功能：获取当前对比度、亮度、饱和度及色调

参数：无。

返回：对比度、亮度、饱和度及色调对应的值

(1.11) int CPE_Grab_VGA(int dispX,int dispY)

功能：实时采集图像至显示卡帧存体，即实时显示。

参数：(dispX,dispY)：图像在屏幕的左上角坐标。

返回：　0　　成功；

−1　　CA-CPE-1000 图像卡尚未初始化；

−3　　不接受分配给图像卡的中断号。

举例：CPE_Grab_VGA(0,0)。

(1.12) int CPE_Grab_Xms(int sFrames,int totalFrames)

功能：实时采集图像至扩展内存。

参数：sFrames：采集起始帧，取值范围：1～最大帧；

totalFrames：总共采集的帧数。

返回：
0 成功；
−1 CA-CPE 1000 图像卡尚未初始化；
−2 采集帧值非法；
−3 不接受分配给图像卡的中断号。

举例：CPE_Grab_Xms (1,1)。

说明：使用该函数可以单帧采集(totalFrames==1)，也可以连续采集(totalFrames>1)。

② 文件管理函数

(2.1) int ReadRaw2XMS(char *fname,int nframe)

功能：读单帧 .RAW 格式图像文件至扩展内存，当函数(1.8)将模式设置为真彩色方式时，读取真彩色 .RAW 文件；模式为灰度方式时，可读取 256 灰度 .RAW 文件。

参数：*fname —文件名；

frame —文件读到第几帧，取值范围：1～最大帧。

返回：
0 成功；
−1 帧值非法；
−3 文件打开失败；
−4 内存分配错误；
−5 内存管理错误。

举例：int nframe=1；

ReadRaw2XMS("TMP.RAW",nframe)。

(2.2) int WriteXMS2Raw(char *fname,int nframe)

功能：将扩展内存某帧图像数据写为单帧 .RAW 格式图像文件，当函数(1.8)将模式设置为真彩色方式时，可保存真彩色 .RAW 文件；模式为灰度方式时，可保存 256 灰度 .RAW 文件。

参数：*fname：文件名；

nframe：文件读到第几帧，取值范围：1～最大帧。

返回：
0 成功；
−1 帧值非法；
−2 磁盘上没有足够的空间；
−3 文件打开失败；
−4 内存分配错误；
−5 内存管理错误。

举例：int nframe=1；

WriteXMS2Raw("TMP.RAW",nframe)。

(2.3) int ReadSrs2XMS(char *fname,int frameNum)

功能：读序列图像文件至扩展内存，根据函数(1.8)的设置，可分别读取真彩色或灰度的图像文件。

参数：*fname：文件名；

frameNum：共读几帧到扩展内存。

返回：
0 成功；
−1 帧值非法；
−3 文件打开失败；
−4 内存分配错误；

－5　　内存管理错误；

＞0　　读到第儿帧出现错误。

举例：int frameNum　3；

ReadSrs2XMS("TMP. SRS",nframe)。

(2.4) int WriteXMS2Srs(char ＊fname,int sframe,int eframe)

功能：从扩展内存写图像信息至序列图像文件，根据函数(1.8)的设置，可分别保存真彩色或灰度的图像文件。

参数：＊fname：文件名；

sframe：从哪一帧开始保存；

eframe：到哪 ·帧结束保存。

返回：　0　　成功；

－1　　帧值非法；

－2　　磁盘上没有足够的空间；

－3　　文件打开失败；

－4　　内存分配错误；

－5　　内存管理错误；

＞0　　写到第儿帧出现错误。

举例：int sframe－1,eframe－5；

WriteXMS2Srs("TMP. SRS",sframe,eframe)。

(2.5) int WriteXMS2Bmp(char ＊szBmpfile,int nframe)

功能：从扩展内存写某帧图像数据为 .BMP 格式图像文件，根据函数(1.8)的设置，分别保存真彩色或灰度图像数据。

参数：＊szBmpfile：文件名；

nframe：保存哪一帧的图像信息，取值范围：1～最大帧。

返回：　0　　成功；

－1　　帧值非法；

－2　　磁盘上没有足够的空间；

－3　　文件打开失败；

－4　　内存分配错误；

－5　　内存管理错误。

举例：int nframe＝1；

WriteXMS2Bmp("TMP. BMP",nframe)。

③ VGA 函数

(3.1) int InitVesa(int wVesaMode)

功能：初始化 VGA。

参数：wVesaMode：图像显示模式，

在真彩色采集方式下支持0x112(640×480×32)及

0x115(800×600×32)模式。

在灰度图像采集方式下支持0x101(640×480×8)及

0x103(800×600×8)模式。

返回：　0　　　　成功；

－1、－2　　VGA 卡不符合 VESA 标准；

－3　　　　不接受的显示模式。

举例：InitVesa(0×112)。

注意：用户在使用本函数库提供的实时显示"函数，或显示图像"等函数之前必须首先调用本函数之后，必须使用函数(3.2)，才能使图形显示模式返回至普通的文本显示状态。

(3.2) int CloseVesa(void)

功能：关闭 VGA。

参数：无。

返回：0　　成功；

　　其他 VGA 卡不符合 VESA 标准；

举例：CloseVesa()；

(3.3) int DispXMS(unsigned int wMode,unsigned int wFrame,\

　　　　　　int start_X,int start_Y)

功能：显示扩展内存中真彩色图像。

参数：wMode：显示模式，0：640×480×32 位模式；

　　　　　　1：800×600×32 位模式；

　　wFrame：显示第几帧图像，取值范围：1～最大帧；

　　(start_X,start_Y)：图像在屏幕上显示位置的左上角坐标。

返回：　0　　成功；

　　−1　　帧值非法；

　　−3　　内存管理错误。

举例：int x=0,y=0,nframe=2；

　　InitVesa(0x112)；

　　DispXMS(0,nframe,x,y)；

　　Getch()；

　　CloseVesa()。

说明：调用本函数前，请确认显示卡已进入 32bit 显示模式。

(3.4) void DISP_640×480(int x,int y,int nframe,\

　　　　　　int Frame_Field_Mark)

功能：在 640×480 分辨率下显示扩展内存中灰度图像。

参数：(x,y)：图像在屏幕上显示位置的左上角坐标；

　　nframe：显示第几帧图像，取值范围：1～Total_Frames；

　　Frame_Field_Mark：0　按帧显示；

　　　　　　1　按场显示。

返回：无。

举例：int x=0, y=0 nframe=2

　　initVeas(0x101)；

　　DISP_640×480(x,y,nframe,0)；

　　getch()；

　　CloseVesa()。

说明：在 640×480 分辨率下要求 VGA 卡至少有 512kB 显示内存。

(3.5) void DISP 800×600(int x,int y,int nframe,\

　　　　　　int Frame Field_Mark)

功能：在 800×600 分辨率下显示扩展内存中灰度图像。

参数：(x,y)：图像在屏幕上显示位置的左上角坐标；

　　nframe：显示第几帧图像，取值范围：1～Total_Frames；

Frame_Field_Mark：0　按帧显示；

1　按场显示。

返回：无。

说明：在 800×600 分辨率下要求 VGA 卡至少有 512kB 显示内存。

④ 数据传输函数

(4.1) int ReadHLine(int nframe,int y,int x1,int x2,\unsigned char far * buff)

功能：将内存中指定帧的某行图像数据读至缓冲区，根据函数(1.8)的设置，读取真彩色或灰度图像数据。

参数：nframe：该行在第几帧，取值范围：1～最大帧；

(x1,y)：该行的起点；

(x2,y)：该行的终点；

* buff：用户提供的数据缓冲区。

返回：　0　　成功；

　　　-1　　帧值非法；

　　　-4　　内存分配错误；

　　　-5　　内存管理错误。

举例：
```
int y=100,x1=100,x2=199;
int nframe=1,length;
unsigned char buff[0X1000];
length=(x2-x1+1) X4L;
ReadHLine(nframe,y,x1,x2,buff)。
```

说明：在真彩色采集设置下，内存中及缓冲区中图像数据排列方式为 BGRα。

注意：请用户分配足够大的缓冲区。该函数不判断缓冲区大小与用户要求数据量是否相符。

(4.2) int WriteHLine(int nframe,int y,int x1,int x2,\
unsigned char far * buff)

功能：将缓冲区中的数据写至内存中指定帧的某行中去，根据函数(1.8)的设置，写入真彩色或灰度图像数据。

参数：nframe：该行在第几帧，取值范围：1～最大帧；

(x1,y)：该行的起点；

(x2,y)：该行的终点；

* buff：用户提供的数据缓冲区。

返回：　0　　成功；

　　　-1　　帧值非法；

　　　-4　　内存分配错误；

　　　-5　　内存管理错误。

举例：
```
int y=100,x1=100,x2=199;
int nframe=1,length;
unsigned char buff[0X1000];
length=(x2-x1+1) X4L;
memset(buff,0X00,length);
WriteHLine(nframe,y,x1,x2,buff)。
```

注意：请用户分配足够大的缓冲区。该函数不判断缓冲区大小与用户要求数据量是否相符。

(4.3) int Readwindow(int nframe,int x1,int y1,int x2,int y2,\
unsigned char far * buff)

功能：将内存中某帧图像的某一矩形部分读到用户缓冲区，根据函数(1.8)的设置，读取真彩色或灰度图像数据。

参数：nframe：矩形部分在第几帧，取值范围：1～最大帧；

(x1,y1)：矩形部分的左上角坐标；

(x2,y2)：矩形部分的右下角坐标；

*buff：用户提供的数据缓冲区。

返回： 0　成功；

-1　要求帧值非法；

-4　内存分配错误；

-5　内存管理错误。

举例：

```
int y1 =100,x1=100,y2=119, x2=199;
int nframe=1,length;
unsigned char buff
length=(y2-y1+1) X(x2-x1+1) X4L;
buff=(unsigned char * )farmalloc(length);
if(buff==NULL{
Close_CPE1000("buff alloc error!!!",0);
  Return ;
  }
ReadWindow(nframe,x1,y1,x2,y2,buff);
Farfree(buff);
```

注意：窗口数据的大小应小于64KB，请用户分配足够大的缓冲区。

(4.4) int Readwindow(int nframe,int x1,int y1,int x2,int y2,\
unsigned char far * buff)

功能：将用户缓冲区中的数据读到内存中某帧图像的某一矩形部分，到用户缓冲区，根据函数(1.8)的设置，写入真彩色或灰度图像数据。

参数：nframe：该矩形部分在第几帧，取值范围：1～最大帧；

(x1,y1)：该矩形部分的左上角坐标；

(x2,y2)：该矩形部分的右下角坐标；

*buff：用户提供的数据缓冲区。

返回： 0　成功；

-1　帧值非法；

-4　内存分配错误；

-5　内存管理错误。

举例：

```
int y1=100,x1=100,y2=119, x2=199;
int nframe=1,length;
unsigned char buff
length=(y2-y1+1)X(x2-x1+1)X4L;
buff=(unsigned char * )farmalloc(length);
if(buff==NULL{
Close_CPE1000"(buff alloc error!!!",0);
  return ;
  }
memset(buff,oxff,length);
```

5

```
WriteWindow(nframe,x1,y1,x2,y2,buff);
farfree(buff);
```

注意：窗口数据的大小应小于 64kB，请用户分配足够大的缓冲区。

(4.5) unsigned long ReadPixel(int nframe,int x,int y)

功能：读像素，根据函数 1.8 的设置，读出真彩色或灰度像素值。

参数：nframe：该像素在第几帧，

取值范围：1～Total_Frames；

(x,y)：像素点坐标；

返回：像素值，对 CA-CPE-1000 图像卡，范围为：0～255。

举例：
```
int pixel,x=100, y -100,nframe=10;
pixel=ReadPixel(nframe,x,y);
```

(4.6) int WritePixel(int nframe,int x,int y,unsigned long val)

功能：写像素，根据函数 1.8 的设置，写入真彩色或灰度像素值。

参数：nframe：该像素在第几帧，

取值范围：1～Tote_Frames；

(x,y)：像素点坐标；

val：像素值，取值范围：0～255；

返回：
0　成功
-1　失败

举例：
```
int result,pixel=128,x=100, y=100,nframe=10;
result=WritePixel(nframe,x,y, pixel);
```

5. Windows 3.x 环境下软件说明

(1) 软件概述

由于 CA-CPE-1000 占用计算机内存存放采集的图像信息，在 Windows 3.x 环境下必须为 CA-CPE-1000 分配一段足够大的连续内存。为此，系统提供一个虚拟驱动程序 VXD.386 进行内存管理。

系统在 Windows 3.x 下提供了演示程序 TCPEW.EXE，该程序演示了系统的采集、回放及文件管理等功能。如仅需采集图像生成文件，可直接使用演示程序。若需要利用 CA-CPE-1000 采集卡嵌入到您所编制的程序中，您可以详细阅读 TCPEW.EXE 的源程序 TCPEW.C，作为系统提供的例子程序。

(2) 内存分配使用说明

请将 VXD.386 拷贝到 Windows 3.x 的 SYSTEM 目录下。

打开 Windows 3.x 的配置文件 SYSTEM.INI，找到[386Enh]节，加入一条语句：

device=vxd.386

在 SYSTEM.INI 中加入新的节，名为[CA-IMAGE]；

在[CA-IMAGE]节中加入一条语句：

FrameSize =xxxxh

其中的'xxxx'表示用户需要为 CA-CPE-1000 采集卡分配多少内存。这个数值应为 16 进制的数据，以 1KB 为单位。'xxxx'后的'h'表示该数值为 16 进制数据。

例：在 SYSTEM.INI 中有如下语句

```
[386Enh]
device=vxd.386
[CA-IMAGE]
FrameSize  1000h
```

那么，在 Windows 启动时，由 VXD.386 读取需要内存的大小"1000h"，并据此为采集卡分配内存。经计算

后可知，实际为采集卡分配的内存大小为

1000h×400h = 4096×1024 = 4MB

在 Windows 退出时，由 VXD.386 自动释放这段分配的内存。

注意：FrameSize 项中的内存大小数据必须以'h'结尾。

(3) 演示程序使用说明

运行\CPEW 目录下的 TCPEW.EXE 程序，首先进入初始化阶段，检查 CA-CPE-1000 的硬件及分配内存等情况。

初始化通过后，即进入集成环境。

系统能够自动根据当前窗口大小调整采集窗口的设置，并采集一帧到内存，显示在屏幕上。

集成环境中演示了文件管理(File)、彩色采集(Grab)、灰度采集(Mono-Grab)、回放(Play)、选择项(Options)、按比例缩放(Scale)及系统信息(Info)等功能。菜单选择可以用鼠标，也可以用热键选择。每个菜单项均有一个字母带有下划线，如文件管理菜单的 File，这个带下划线的字母就是该菜单的热键，即组合键"Alt + F"。

① 文件管理(File)菜单包括读写单帧图像文件、读写序列图像文件、读写.BMP 格式文件、单帧图像格式的文件转换为 .BMP 格式文件及退出集成环境。

在保存单帧图像文件(Save single file)时，写入磁盘的是当前屏幕上显示的那帧。读单帧图像文件(Open single file)时，将文件内容读到当前帧所在的内存位置并显示出来。

注意：读单帧图像文件后采集参数可能会有所改变。

在保存序列图像文件(Save series file)时，写入磁盘的是当前采集并存在于内存中的全部图像。读序列图像文件(Open series file)时，依次将序列图像放入扩展内存，若扩展内存不够容纳所有这些图像数据，则会有提示信息。读、写完序列图像文件后会有提示，表示操作已经完成。

注意：读序列图像文件后采集参数可能会有所改变。

保存.BMP 格式的图像文件(Save single to .BMP)，写入磁盘的是当前屏幕上显示的那帧，按照.BMP 格式保存。可以在任何其他可接受.BMP 格式的程序中读出。读.BMP 格式的图像文件(Open .BMP file)，仅能读出 24bit 的.BMP 格式图像文件或 8bit 带查找表的.BMP 格式图像文件。

格式转换(Transform.RAW to.BMP)提供了由.RAW 格式文件转换到.BMP 格式文件的功能。

② 采集功能包括采集单帧图像、采集序列图像到扩展内存，以及实时显示。

注意：当选择项(Option)菜单选中真彩色采集方式时，本菜单才有效。

采集单帧图像(Grab single frame)是采集一帧图像到扩展内存后显示在屏幕上，并持续下去，直至有键盘键按下或有鼠标键按下。

实时采集序列图像到扩展内存(Grab real-time(XMS))，是采集一系列图像至扩展内存并将最后一帧显示在屏幕上。总共可采集帧数受占有内存的大小及采集窗口的大小所限，可以通过系统信息(Info)选项得到总采集帧数。采集后的图像演播可利用回放(Play)功能。

注意：在序列采集过程中屏幕静止不动，直到采集完毕显示最后一帧图像。

实时采集动态图像到 VGA(Grab real-time(VGA))，是采集图像并实时显示。有键盘或鼠标键按下则停止采集，恢复至 Windows 正常状态。

注意：仅在 Windows 显示模式为真彩色时才能演示该功能。

单帧采集动态图像到 VGA(Single frame to VGA)，是采集一帧图像到显示卡，每采集完一帧图像后即恢复至 Windows 正常状态。若有键盘或鼠标键按下则停止采集；若无上述消息，则持续上述过程。

注意：仅在 Windows 显示模式为真彩色时才能演示该功能。

③ 灰度采集功能包括采集单帧图像和采集序列图像到扩展内存以及实时显示动态图像。

注意：当选择项菜单选中灰度采集方式时，本菜单才有效。

采集单帧图像(Grab single frame)是采集一帧图像到扩展内存后显示在屏幕上，并持续下去，直至有键盘键按下或有鼠标键按下。

实时采集序列图像到扩展内存(Grab real-time(XMS)，是采集一系列图像至扩展内存并将最后一帧显示在屏幕上。总共可采集帧数受占有内存的大小及采集窗口的大小所限，可以通过系统信息(Info)选项得到总采集帧数。采集后的图像演播可利用回放(play)功能。

注意：在序列采集过程中屏幕静止不动，直到采集完毕显示最后一帧图像。

实时采集动态图像到 VGA(Grab real-time(VGA))，是采集图像并实时显示，直至有键盘或鼠标键按下停止采集，恢复至 Windows 正常状态。

注意：仅在 Windows 显示模式为 256 色时才能演示该功能。

单帧采集动态图像到 VGA(Single frame to VGA)，是采集一帧图像到显示卡，每采集完一帧图像后即恢复至 Windows 正常状态。若有键盘或鼠标键按下则停止采集；若无上述消息，则持续上述过程。

注意：仅在 Windows 显示模式为 256 色时才能演示该功能。

回放功能包括回放、向前进一帧、向后退一帧及帧/场显示。帧/场显示是指图像显示可以按帧方式显示也可以按场方式显示。

④ 回放(Play)功能是将当前存放在内存中的各帧图像逐个显示。

向前进一帧(Play forward)是显示当前帧的前一帧。

向后退一帧(Play backward)是显示当前帧的下一帧。

帧/场显示是个单项选择，二者必居其一。一帧完整的图像由两场组成，按帧方式显示(Display frame)是将这完整的两场显示出来；按场方式显示(Display field)是将一场的图像扩展为一帧，这样可以消除两场之间的时间差造成的错位。

⑤ 选择项功能包括设置采集模式、设置输入信号源(Video)、设置亮度(Brightness)、对比度(Contrast)、饱和度(Saturate)和色调(Hue)等。

选择 True Color 640×480 或 True Color 800×600 模式，则采集菜单(Grab 中的各项有效。但是 Windows 3.x 当前的显示模式应为 32 位真彩色，否则无法进行动态实时显示。

选择 Grayscale 640×480 或 Grayscale 800×600 模式，则灰度采集菜单(Mono-Grab)中的各项有效。但是 Windows 3.x 当前的显示模式应为 256 色模式，否则无法进行动态实时显示。

输入信号可以在 6 路中任选一路。若选择的输入源有信号输入，则采集可以得到图像信号，否则是全黑信号。在一般情况下，提供的黄色信号线是一号输入源，为缺省设置。

⑥ 按比例缩小(Scale)功能是设置采集图像比例。

按比例缩放功能根据菜单的不同选项设置采集时的缩小比例，100%情况下表示按原来大小采集。在缩小采集的情况下，采集视野不变，采集过程中硬件自动根据缩小比例抽取图像信息，得到按比例缩小的图像。

⑦ 信息显示功能汇集有关 CA-CPE-1000 的当前参数，并显示出来。内容包括：CA-CPE-1000 占用的中断号、共有多少 KB 的内存分配给 CA-CPE-1000，当前采集窗口大小，目前最多能存放多少帧图像，多少帧中已有图像，当前显示的是第几帧，该帧在内存中的起始位置等信息。按"确定"键后返回主菜单。

(4) 函数库详解

CA-CPE-1000 的 Windows 3.x 环境下的函数库包括三类函数，分别是系统控制函数、文件管理函数和数据传输函数。

① 系统控制函数

A 类. 初始化函数

(1.1) 初始化 CA-CPE-1000 图像采集卡

DWORD FAR PASCAL Init_CPE1000(void)

(1.2) 关闭 CA-CPE-1000 图像采集卡

int FAR PASCAL Close_CPE1000(void)

(1.3) 获取 CA-CPE-1000 所在 PCI 插槽的中断号

int FAR PASCAL Get_CPE1000IRQnum(void)

B 类. 设置函数

(1.4) 设置采集图像的位置及大小

库函数请以 USERCPEW.H 头文件中的定义为准

int FAR PASCAL Set_Grab_window(int,int,int,int)

(1.5) 设置 x 方向上及 y 方向上的采集起始点

int FAR PASCAL Set_StartPoint(int,int)

(1.6) 设置图像显示方式

VOID FAR PASCAL Set_Display_Mode(BOOL)

(1.7) 调整图像采集对比度、亮度、色调及饱和度

int FAR PASCAL Modify_Contrast(BYTE)

int FAR PASCAL Modify_Brightness(BYTE)

int FAR PASCAL Modify_Hue(BYTE)

int FAR PASCAL Modify_Saturate(BYTE)

(1.8) 选择输入信号源

VOID FAR PASCAL Set_Input_Ssource(nit)

(1.9) 设置采集窗口缩小比例

int FAR PASCAL Set_Zoom_value(int,int,WORD,WORD)

(1.10) 指示当前 Windows 显示模式

VOID FAR PASCAL Set_VGA_Mode(BYTE)

(1.11) 设置采集方式、真彩色或灰度图像

int FAR PASCAL Set_Pixel_Format(WORD)

(1.12) 得到扩展内存中某帧的物理起始地址

DWORD FAR PASCAL Get_Frame_Address(WORD)

(1.13) 设置采集频率

VOID FAR PASCAL Set_Grab_Freq(WORD)

C 类.采集函数

(1.14) 实时采集图像至扩展内存

int FAR PASCAL CPE_Grab_Xms(int,int)

(1.15) 启动动态显示图像

int FAR PASCAL CPE_RealGrab_VGA(WORD,WORD)

(1.16) 停止动态显示图像

int FAR PASCAL CPE_Exit_Grab_VGA(WORD,WORD)

(1.17) 单帧动态显示图像

int FAR PASCAL CPE_NormalGrab_VGA(WORD,WORD)

② 文件管理函数

(2.1) 读.RAW 格式单帧图像文件至扩展内存

int FAR PASCAL ReadRaw2XMS(LPSTR,WORD)

int FAR PASCAL ReadRaw2XMS_MONO(LPSTR,WORD)

(2.2) 写.RAW 格式单帧图像文件

int FAR PASCAL WriteXMS2Raw(LPSTR,WORD)

int FAR PASCAL WriteXMS2Rawar_MONO(LPSTR,WORD)

(2.3) 读序列图像文件至扩展内存

int FAR PASCAL ReadSrs2XMS(LPSTR)

(2.4) 写序列图像文件

int FAR PASCAL WriteXMS2Srs(LPSTR,WORD,WORD)

(2.5) 写.BMP 格式图像文件

int FAR PASCAL WriteXMS2Bmp(LPSTR,WORD)

int FAR PASCAL WriteXMS2Bmp_MONO(LPSTR,WORD)

(2.6) 将.BMP 格式图像读入扩展内存

int FAR PASCAL ReadBmp2XMS(LPSTR,WORD)

(2.7) .RAW 格式文件转换到.BMP 格式文件

int FAR PASCAL Raw2Bmp(LPSTR,LPSTR)

(2.8) 获取 .RAW 格式文件宽度

WORD FAR PASCAL GetMpeFileWidth(void)

(2.9) 获取 .RAW 格式文件高度

WORD FAR PASCAL GetMpeFileHeight(void)

③ 数据传输函数

(3.1) 读某帧图像至缓冲区

int FAR PASCAL ReadXMS2Buf(WORD,BYTE huge *);

int FAR PASCAL ReadXMS2Buf_MONO\(WORD,BYTE huge *);

(3.2) 读某帧图像至缓冲区,图像数据未经倒置转换

int FAR PASCAL ReadXMS2Buf_NoReverse\(WORD,BYTE huge *);

(3.3) 写缓冲区中的图像至某帧

int FAR PASCAL WriteBuf2XMS(WORD, BYTE huge *);

int FAR PASCAL WriteBuf2XMS_MONO\(WORD,BYTE huge *);

(3.4) 将显示在屏幕上的某一部分图像写至扩展内存

int FAR PASCAL WriteVGA2XMS(WORD,WORD,\WORD,WORD)

下面详细解释各个函数:

① 系统控制函数

(1.1) DWORD FAR PASCAL Init_CPE1000(void)

功能:初始化 CA-CPE-1000 图像采集卡。

参数:无。

返回:返回的值为分配给 CA-CPE-1000 的内存大小,以 kB 为单位。

0X00xxxxxxL 成功,返回值为内存大小;

0X81000000L Vxd.386 没有正确安装;

0X84000000L 没有找到配置文件 RSVS.XFG;

0X85xxxxxxL 视频硬件不匹配,其中后 6 个字节为内存大小;

0X86xxxxxxL 不能进行实时显示,其后 6 个字节为内存大小;

0X87000000L 没有插卡;

错误码为 0X85xxxxxxL 或 0X86xxxxxxL 时,不能进行实时显示,但可以采集图像到内存。

举例:DWORD dwMemSize;

dwMemSize=inti_CPE1000();

注意:在使用其他 CPE 1000 图像卡库存函数之前,必须调用该函数。

(1.2) int FAR PASCAL Close_CPE1000(void)

功能:关闭 CA-CPE-1000 图像采集卡。

参数:无。

返回:0　　　　成功;

其他　　失败。

注意:该函数必须与函数 1.1 配套使用。

(1.3) int FAR PASCAL Get_CPE1000-IRQnum(void)

功能：获取 CA-CPE-1000 所在 PCI 插槽的中断号。

参数：无。

返回：CA-CPE-1000 所在 PCI 插槽的中断号。

(1.4) int FAR PASCAL Set_Grab_window (int x1,int y1,\int x2,int y2)

功能：设置采集图像的位置与大小。

参数：(x1,y1)：采集图像的左上角坐标；

(x2,y2)：采集图像的右下角坐标。

实际采集窗口大小：

宽度 =x2−x1+1；高度 =y2−y1+1；

返回：返回值为在该设置情况下，最大可采集的帧数。

>0　成功，最大可采集帧数。

−1　高度或宽度的设置值为零；

−2　高度或宽度的设置值超出最大值；

−3　内存操作错误；

举例：int Total_Frames；

Total_Frames=Set_Grab_Window(0,0,639,479)；

例中设置的实际采集窗口宽度和高度为 640×480。

说明：此偏移位置的设置表示在摄像头视野中的采集范围。

(1.5) int FAR PASCAL Set_StartPoint(int StartX0,int StartY0)

功能：设置 x 方向及 y 方向上的采集起始点，一般用户不必另行设置。

参数：StartX0：x 方向上采集起始点，缺省值为 0×12；

StartY0：y 方向上采集起始点，缺省值为 0×28。

返回：0　成功。

说明：如图 5 中所示，在 x 方向上摄像头全部信号包括行同步信号、行消隐信号及可视的图像信号等。若 x 方向上的起始点选择得较小，则有可能采集到行消隐信号，现象是图像的左侧有黑条；若双方向上的起始点选择得过大，则有可能找不到行同步信号，出现图像扭曲等错误现象。因此，在用户调整这一参数时，请注意正确设置。

同理，在 y 方向上摄像头全部信号包括场同步信号、场消隐信号及可视图像信号等。若 y 方向上的起始点选择得较小，则有可能采集到场消隐信号，现象是图像的上部有黑条；若选择得过大，也会出现图像信号不正常的现象。

(1.6) VOID FAR PASCAL Set_Display_Mode(BOOL bMark)

功能：设置图像显示方式(按帧或场)。

参数：bMark：TRUE 按帧显示，

FALSE 按场显示。

返回：无。

举例：Set_Display_Mode(TRUE)。

(1.7) int FAR PASCAL Modify_Contrast(BYTE contrast)；

nit FAR PASCAL Modify_Brightness(BYTE brightness)；

nit FAR PASCAL Modify_Hue(BYTE hue)；

nit FAR PASCAL Modify_Saturate(BYTE saturate)

功能：调整图像采集对比度、亮度、色调及饱和度等，

参数：对比度、亮度、色调及他和度参数值，取值范围：0～255。

返回：0　成功。

(1.8) void FAR PASCAL Set_Input_Source(int vin_select)

功能：选择输入信号源。

参数：vin_select：信号源选择，取值范围：0～8。

返回：无。

举例：Set_Input_Source(0)；

说明：取值　　信号源

　　0～5　　CVBS No.1～6 输入

　　6　　S-Video Y：No.1 C：No.6

　　7　　S-Video Y：No.3 C：No.2

　　8　　S-Video Y：No.5 C：No.4

　　No.1～6 指输入信号线的编号。

(1.9) int FAR PASCAL Set_Zoom_Value(int scaleX,\
　　int scaleY,WORD *wBackWidth,WORD *wBackHeight)

功能：设置采集窗口缩小比例。

参数：scaleX：图像采集 x 方向上的缩小比例，以百分比(%)为单位，取值范围：1～100；

　　scaleY：图像采集 y 方向上的缩小比例，以百分比(%)为单位，取值范围：1～100；

　　wBackWidth：根据图像采集 x 方向上的缩小比例及采集窗口原来的大小，计算出的缩小后窗口 x 方向上的宽度。

　　wBackHeight：根据图像采集 y 方向上的缩小比例及采集窗口原来的大小，计算出的缩小后窗口 y 方向上的高度。

返回：返回值为缩小采集窗口后的最大采集帧数。

　　>0　　成功，最大采集帧数；

　　-1、-2　　x 方向或 y 方向上缩小比例设置不当。

举例：

```
WORD wTotalFrames;
BOOL bZOOmMark=FALSE;
WORD wTempWidth,wTempHeight
WTotalFrames=Set_Zoom_Value(50,90),\
                &wTempWidth,&wTempHeight);
bZoomMark=TRUE;
```

　　设置采集窗口缩小比例，x 方向上为 50%，y 方向上为 90%。

说明：设置采集缩小比例后，采集窗口缩小，但采集视野不变。

(1.10) VOID FAR PASCAL Set_VGA_Mode(BYTE bVgamode)

功能：用于与以前的软件兼容，现为空函数。

(1.11) int FAR PASCAL Set_Pixel_Format(WORD p_format)

功能：设置采集方式，真彩色或灰度图像。

参数：P_format：采集方式，0X00　　真彩色采集方式；

　　　　　　　　0XC3　　256 级灰度采集方式。

返回：返回值为在该设置情况下，最大可采集帧数。

　　>0 成功，最大可采集帧数。

举例：

```
#define RGB24——PACKED 0X00
int Total_Frames;
Total_Frames=Set_Pixel_Format(RGB24_PACKED)。
```

说明：系统初始化时，默认设置为真彩色采集方式。

注意：真彩色采集方式下一个像素由 32bit 体现，数据排列方式为"蓝绿红"；灰度采集方式下一个像素由

8bit 体现。

(1.12) DWORD FAR PASCAL Get_Frame_Address\
(WORD wFrame)

功能：得到扩展内存中某帧的物理起始地址。

参数：wFrame：帧号，取值范围：1～最大帧。

返回：返回值即为该帧在扩展内存中的物理起始地址。

0X00L 帧值非法。

(1.13) VOID FAR PASCAL Set_Grab_Freq(WORD wFreq)

功能：设置采集频率。

参数：wFreq：图像采集时，几帧中采集一帧。

返回：无。

说明：采集频率的设置，对序列采集（采到内存或 VGA）有效。

(1.14) int FAR PASCAL CPE_Grab_Xms(int sFrames，\
int totalFrames)

功能：实时采集图像至扩展内存。

参数：sFrames：采集起始帧，取值范围：1～最大帧；

totalFrames：总共采集的帧数。

返回： 0 成功；

－1 CA-CPE-1000 图像卡尚未初始化；

－2 采集帧值非法；

－3 不接受分配给图像卡的中断号。

举例：CPE_Grab_Xms(1,1)。

说明：使用该函数可以单帧采集(totalFrames ＝ －1)，也可以序列采集(totalFrames＞1)。

(1.15) int FAR PASCAL CPE_RealGrab_VGA(WORD dX0,\WORD dY0)

功能：启动动态显示图像.

参数：(dX0,dY0)：图像显示在屏幕的左上角坐标（绝对坐标）。

返回： 0 成功；

－1 CPE1000 图像采集卡尚未初始化；

－2 当前视频显示模式不匹配；

－3 视频硬件不匹配。

举例：见函数(1.16)"停止动态显示图像"中的示例。

(1.16) int FAR PASCAL CPE_Exit_Grab_VGA(WORD dX0,\WORD dY0)

功能：停止动态显示图像。

参数：(dX0,dY0)：图像在屏幕上显示的左上角坐标（绝对坐标）。

返回： 0 成功；

－1 CA-CPE-1000 图像采集卡尚未初始化；

－2 图像卡没有进行实时显示。

举例：bGrabMark＝TRUE

```
// 开始动态显示
if( CPE_RealGrab_VGA(winOrgPoint.x,winOrgPoint.y)< 0)
  return 0;
while(bGrabMark == TRUE){
  //截取窗口消息
```

```
    PeekMessage(&msg,NULL,0,0,PM_REMOVE);
    //若有键按下,设置停止采集标志
    if(msg.message == WM_KEYDOWN)
            bGrabMark = FALSE;
  }// end of(bGrabMark)
CPE_Exit Grab_VGA(winOrgPoint.x, winOrgPoint.y)。
```

注意：此函数必须与函数(1.15)"启动动态显示"配套使用。

(1.17) int FAR PASCAL CPE_NormalGrab_VGA(WORD dX0,\WORD dY0)

功能：单帧动态显示图像。

参数：(dX0,dY0)：图像显示在屏幕的左上角坐标(绝对坐标)。

返回： 0 成功；

- −1 CA-CPE-1000 图像采集卡尚未初始化；
- −2 当前视频显示模式不匹配；
- −3 不接受分配给图像卡的中断号；
- −4 视频硬件不匹配。

举例：
```
bGrabMark= TRUE;
    while(bGrabMark == TRUE){
    //截取窗口消息
    PeekMessage(&msg,NULL,0,0,PM_REMOVE);
    //若有键按下,设置停止采集标志
    if(msg.message==WM_KEYDOWN)
        bGrabMark==FALSE;
    //实时采集并显示一帧
if(CPE_NomalGrab_VGA(winOrgPoint.x,winOrgPoint.y)< 0)
    bGrabMark = FALSE;
    }// end of(bGrabMark)。
```

说明：此函数实时采集并显示一帧图像。与(1.15)函数不同的功能在于：(1.15)函数的功能是启动实时显示,之后只有通过调用(1.16)函数系统才能恢复至图像采集前的状态。因此(1.15)函数与(1.16)函数必须配套使用。而此函数是一个完整状态的函数,函数返回后,系统即恢复至图像采集前的状态。请用户根据自己的需要来调用(1.15)或(1.17)函数。

② 文件管理函数

(2.1) int FAR PASCAL ReadRaw2XMS(LPSTR szFileName,\
WORD wFrame)

int FAR PASCAL ReadRaw2XMS_MONO\
(LPSTR szFileName,WORD wFrame)

功能：读单帧.RAW 格式图像文件至扩展内存。第一个函数读取真彩色.RAW 文件;第二个函数读取 256 灰度.RAW 文件。

参数：szFileName：文件名；

wFrame：文件读到第几帧,取值范围：1～最大帧。

返回： 0 成功；

- −1 帧值非法；
- −3 文件打开失败；
- −4 内存分配错误；
- −5 文件读错误。

举例：intnframe＝1

ReadRaw2XMS("TMP.RAW",nframe)。

(2.2) int FAR PASCAL WriteXMS2Raw(LPSTR szFileName,\WORD wFrame)；

int FAR PASCAL WriteXMS2Raw_MONO\(LPSTR szFileName,WORD wFrame)

功能：将扩展内存某帧图像信息写为单帧.RAW格式图像文件。第一个函数写真彩色.RAW文件；第二个函数写256灰度.RAW文件。

参数：szFileName：文件名；

wFrame：文件读到第几帧，取值范围：1～最大帧。

返回：0　成功；

－1　帧值非法；

－2　磁盘上没有足够的空间；

－3　文件打开失败；

－4　内存分配错误；

－5　文件写错误。

举例：int nframe＝1

WriteXMS2Raw("TMP.RAW",nframe)。

(2.3) int FAR PASCAL ReadSrs2XMS(LPSTR szFileName)

功能：读序列图像文件至扩展内存。

参数：szFileName：文件名。

返回：－2　文件打开失败；

－3　内存分配错误；

－4　文件读错误；

其他从该序列文件中读出的帧数。

举例：ReadSrs2XMS("TMP .SRS")。

(2.4) int FAR PASCAL WriteXMS2Srs(LPSTR szFileName,\

WORD wSframe,WORD wEframe)

功能：从扩展内存写图像信息为序列图像文件。

参数：szFileName：文件名；

wSframe：从哪一帧开始保存；

wEframe：到哪一帧结束保存。

返回：0　成功；

－1　帧值非法；

－2　磁盘上没有足够的空间；

－3　文件打开失败；

－4　为内存分配错误；

－5　文件写错误。

举例：int sframe ＝ 1　eframe＝5

WriteXMS2Srs("TMP.SRS",sframe, eframe)。

(2.5) int FAR PASCAL WriteXMS2Bmp(LPSTR siFileName,\WORD wFrame)

int FAR PASCAL WriteXMS2BmpMONO\

(LPSTR szFileName,WORD wFrame)；

功能：从扩展内存写某帧图像信息为.BMP格式图像文件.第一个函数写真彩色.RAW文件；第二个函数写256灰度.RAW文件。

参数：szFileName：文件名；

wFrame：保存哪一帧的图像信息。

返回： 0 成功；

-1 帧值非法；

-2 磁盘上没有足够的空间；

-3 文件打开失败；

-4 内存分配错误；

-5 文件写错误。

举例：int nframe = 1

WriteXMS2Bmp("TMP.BMP",nframe)。

(2.6) int FAR PASCAL ReadBmp2XMS(LPSTR szFileName,\WORD wFrame)

功能：将.BMP格式图像读入扩展内存。

参数：szFileName：文件名；

wFrame：文件读到第几帧，取值范围：1～最大帧。

返回： 0 成功；

-1 帧值非法；

-2 文件打开失败；

-3 内存分配错误；

-4 文件读错误。

举例：int nframe = 1

ReadBmp2XMS("TMP.RAW",nframe)，

说明：本函数只能正确读取24bit真彩色图像及8bit带查找表的灰度图像.

(2.7) int FAR PASCAL Raw2Bmp(LPSTR szRawfile,\LPSTR szBmpfile)

功能：.RAW格式文件转换到.BMP格式文件

参数：szRawfile：RAW格式文件的文件名；

szBmpfile：BMP格式文件的文件名。

返回： 0 成功；

-1 文件打开失败；

-2 内存分配错误。

-3 .RAW文件读错误；

-4 .BMP文件写错误。

(2.8) WORD FAR PASCAL GetMpeFileWidth(void)

功能：获取.RAW格式文件宽度。

参数：无。

返回：返回值为.RAW格式文件宽度。

举例：WORD wRawHeight；

wRawHeight=GetMpeFileHeight()。

(2.9) WORD FAR PASCAL GetMpeFileHeipht(void)

功能：获取.RAW格式文件高度。

参数：无，

返回：返回值为.RAW格式文件高度。

举例：WORD wRawHeight

wRawHeight = GetMPeFileHeight()。

(3) 数据传输函数

(3.1) int FAR PASCAL ReadXMS2Buf(WORD wFrame,\BYTE huge * lpXMS)

int FAR PASCAL ReadXMS2Buf_MONO(WORD wFrame,\
BYTE huge *lpXMS)

功能：读内存中某帧图像至缓冲区，第一个函数读真彩色图像，第二个函数读256灰度图像。

参数：wFrame：读第几帧，取值范围：1~最大帧；

lpXMS：数据缓冲区，由用户分配。

返回：0　成功；

－1　帧值非法；

－2　内存分配错误。

举例：
```
int nframe= 1;
BYTE huge * lpDib;
HGLOBAL      hglb;
DWORD        dwDibSize;
DwDibSize =(DWORD)WimgWidth × WimgHeight×3L;
hglb = GlobalAlloc(GMEM_MOVEABLE,dwDibsize);
lpDib = (BYTE huge * )GlobalLock(hglb);
ReadxMS2Buf(nframe,lpDib);
GlobalUnLock(hglb);
GlobalFree(hglb)。
```

说明：该函数将用户指定的某帧图像数据放到用户分配的缓冲区中。缓冲区中的图像数据格式为一个像素占用3个字节，分别为"蓝绿红"信息。

注意：请用户分配足够大的缓冲区，该函数不判断缓冲区的大小与用户要求的数据量是否相符。

(3.2) int FAR PASCAL ReadXMS2Buf_NoRverse\
(WORD wFrame,BYTE huge * lpXMS)

功能：读内存中某帧真彩色图像至缓冲区，此函数中图像数据在从扩展内存传输至缓冲区中时，未经"倒置"处理。

参数：wFrame：读第几帧，取值范围：1~最大帧；

lpXMS：数据缓冲区，由用户分配。

返回：0　成功；

－1　要求帧值非法；

－2　内存分配错误。

说明：该函数将用户指定的某帧图像数据放到用户分配的缓冲区中。缓冲区中的图像数据格式为一个像素占用3个字节，分别为"蓝绿红"信息。此函数与函数(3.1)的不同之处在于，函数(3.1)在数据传递过程中加入了"上下颠倒"操作，这是为了调用GDI函数显示正常的图像。

注意：请用户分配足够大的缓冲区，该函数不判断缓冲区的大小与用户要求的数据量是否相符。

(3.3) int FAR PASCAL WriteBuf2XMS(WORD wFrame,\
BYTE huge * lpDib)

int FAR PASCAL WriteBuf2XMS_MONO(WORD wFrame,\
BYTE huge * lpDib)

功能：第一个函数是将缓冲区中真彩色的图像写至内存中某帧的位置；第二个函数是将缓冲区中256灰度的图像写至内存中某帧的位置。

参数：wFrame：读第几帧，取值范围：1~最大帧；

lpDib：数据缓冲区，由用户分配。

返回：0　成功；

－1　帧值非法；

－2　　内存分配错误。

举例：WriteBuf2XMS(2,lpDib)

假定 lpDib 中的数据是从第一帧读来的数据，上述操作的意义是将第一帧的数据写到第二帧。

注意：请用户注意保持当前采集窗口的大小与该缓冲区的大小一致，否则会导致错误。

(3.1) int FAR PASCAL WriteVGA2XMS(WORD x0,WORD y0,\
WORD dx,WORD dy,WORD wFrame)

功能：将显示在显示器上某一矩形部分的真彩色图像数据写入扩展内存。

参数：(x0,y0)：显示器上矩形区左上角绝对坐标；

dx：需要写入扩展内存的矩形区宽度；

dy：需要写入扩展内存的矩形区长度。

返回：　0　　成功；

－1　　帧值非法；

－4　　失败。

举例：CPE_NormalGrab_VGA(winOrgPoint.x,winOrgPoint.y)

WriteVGA2XMS(winOrgPoint.x,winOrgPoint.y,\

wVGAWidth, wVGAHeight, wCurrent_Frame)。

说明：本函数的作用是将采集到 VGA 上的图像信息保存到内存中。

6. Windows 95 环境下驱动软件包的使用说明

(1) 简介

为 Windows 3.x 设计的 16 位动态连接库存和演示程序可以在 Windows 95 环境下正常工作，但该编程接口只能支持 16 位 Windows 应用程序的开发，而不能支持 Windows 95 下的 Win32API，使用户受到诸多限制。

为此，本公司特别开发了支持 Windows 95 环境下 32 位编程的驱动软件包，该软件包，尽量保持与原 Windows 3.x 下开发软件的兼容性，用户可以很方便地把在原 Windows 3.x 下开发的软件移植到 Windows 95 下，享受 32 位编程的优越性。

(2) 初始设置

将 Vxd.vxd 拷贝到 Windows 95 的 SYSTEM 子目录下。

修改 Windows 95 的 SYSTEM.INI 文件，在[386Enh]一节中加入：

Device＝vxd.vxd

并创建一个新节[CA-Image]，在其中加入

FrameSize＝800h

其中 800h 指分配 2MB 内存作为 CPE1000 的缓存，它是在主机为 16MB 内存时的推荐值。

为方便起见，可将 COEW16.DLL 和 CPEW32.DLL 拷贝到 Windows 95 的 SYSTEM 子目录下。请将 Windows 95 的显示模式设置到 32 位真彩色，这时具有较好的性能。

(3) 编程

提供的库存函数与 Windows 下软件包所用库存函数基本相同，区别如下：

函数名前均以“W32_”打头。

原来使用 huge 指针的地方均改为普通指针。

函数的具体用法请参考 Windows 3.x 下软件包中无“W32_”打头的同名库函数的说明。

为正确使用这些函数，请在函数调用之前包含头文件 CPEW32.H。

在将原来 Windows 3.x 下开发的软件移植到 Windows 95 时，有时需要使用强制类型转化解决移植时产生的数据类型不一致问题。

参考文献

[1] R.C.冈萨雷斯,P.温茨著,李叔梁等译,数字图像处理,科学出版社,1982.

[2] A.罗森菲尔德,A.C.卡克著,余英林等译,数字图像处理,人民邮电出版社,1982.

[3] 本肇编著,画像の情报处理,コロナ社,1978.

[4] 宫川洋,渡部叡编著,画像エレトロニクスの基础,コロナ社,1975.

[5] W.K.普拉特著,高荣坤译,数字图像处理学,科学出版社,1984.

[6] 习鸟,画像特徵テレビ誌29卷,第二号,1975.

[7] 藤尾,画像量との视觉系,信学誌,59卷,11号,1976.

[8] H. H. Hutson, Color Television Theory PAL-system Principles and Receiver Circuitry, Company Limited, 1971.

[9] W.K.Pratt, Spatial Transform Coding of Color Images, IEEE Trans. Commun. Tech., Com-19,12, Desember 1971.

[10] E.R.Kretzmer, Statistics of Television Signals, The Bell System Technical Journal, 1952.

[11] W.D.斯坦利著,常迥译,数字信号处理,科学出版社,1978.

[12] 蒋先进编,微光电视,国防工业出版社,1984.

[13] 木户出,正继,筱田,英苑,快速数字图像处理装置的发展趋势,国外自动化科学技术出版社,1979.

[14] A Gilbert, A Real-Time Video Tracking System, IEEE Trans. PAMI, Vol.2, No.1:47~56,1980.

[15] 袁保宗编著,数字信号处理与通信,人民邮电出版社,1981.

[16] J.W.Cooley, J.W.Tukey, An Algorithm for Machine Calculation of Complex Fourier Series, Math. Computation, 1965.

[17] 胡征,樊昌信编著,沃尔什变换及其在通信中的应用,人民邮电出版社,1980.

[18] E.L.Hall, Computer Image Processing and Pattern Recognition, Academic Press, 1979.

[19] W.M. Goodall, Television by Pulse Code Modulation, The Bell System Technical Journal, 1951.

[20] A. Review, Picture Coding , Proceedings of IEEE, Vol.68, No.3, 1980.

[21] 张宏基编著,信源编码,人民邮电出版社,1980.

[22] P.A.Wintz, Transform Picture Coding, Proceedings of the IEEE, Vol.60, No.7, 1972.

[23] N. Ahmed, T. Natarajan, K. R. Rao, An Image Processing and Discrete Cosine Transform ,IEEE Trans. Computers, Vol.C-23, 1, 1974.

[24] A. Habibi, Two-dimension Bayesian Estimate of Images, Proceedings of the IEEE Vol.60, No.7, 1972.

[25] K.S.Fu, Digital Pattern Recognition, Springer-Verlag, Berlin Heidelberg New York, 1976.

[26] G.J.Vanderbrug, Azriel Rosenfeld, Two-Stage Template Matching, IEEE Trans. Computers, Vol. C-26, No.4, 1977.

[27] 陈尚勤,魏鸿俊,模式识别理论及应用,成都电讯工程学院出版社,1983.

[28] 付京孙著,戴汝为,胡启恒译,模式识别及其应用,科学出版社,1983.

[29] 李月景编著,图像识别技术及应用,机械工业出版社,1983.

[30] 阮秋琦,视频图像实时的Walsh-Hadarmada变换编码研究,北方交通大学,硕士论文,1981.

[31] 张丽英,视频信号的自适应增量编码装置,北方交通大学,硕士论文,1982.

[32] 楼世博,金晓龙,模糊数学及其应用,国外自动化,2卷,2期,1980.

[33] L.N.Kanal, Interactive Pattern Analysis and Classification System: A Survey and Commentary, Proceedings of the IEEE, Vol.60, No.10, 1972.

[34] J. R. Ulliman, Picture Recognition and Analysis. The Radio and Electronic Engineer, Vol. 47. No. 1, 1977.
[35] K. Preston, A Comparission of Analog and Digital Techniques for Pattern Recognition , Proceedings of the IEEE, Vol. 60, No. 10, 1972.
[36] N. E. Nahi, C. A. Franco, Recursive Image Enhancement Vector Processing, IEEE Trans. Communication, Vol. C-21, 1973.
[37] 阮秋琦,袁保宗,全电视信号实时编码研究,全国信号处理学术会议全国图像处理学术会议论文集,1980.
[38] 阮秋琦,全电视信号实时 Walsh-Hadamard 变换编码研究,全国图像处理学术会议全国图像处理学术会议论文集,1983.
[39] 阮秋琦,图像的伪彩色增强研究,全国图像处理学术会议论文集,1986.
[40] 阮秋琦,模糊图像的恢复研究,全国计算机用户协会学术会议论文集,1986.
[41] F. Mokhtarian, A. Mackworth, Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes, IEEE Trans. PAMI, Vol. 8, No. 1:34~43 1986.
[42] 阮秋琦编著,数字图像处理基础,中国铁道出版社,1988.
[43] Qiuqi Ruan, An Improvement of the Realization Method for the Restoration of a Motion Blurred Image, New Products Show Conference, Chicago, 1987.
[44] Qiuqi Ruan, Dimensioning Inspection on X-ray Images, GE Report, U. S. A. ,1988.
[45] Qiuqi Ruan, The Displacement Measurement in the X-ray Image Using Hierarchical Correlation Method, GE Report, U. S. A. ,1988.
[46] Qiuqi Ruan, PDP-11 image processing system, AI&CV LAB Report, U. S. A. ,1989.
[47] Qiuqi Ruan, Intelligent Transtent fault monitor. Lead Ltd. Report, U. S. A. ,1989.
[48] 阮秋琦,用分层相关法实现运动目标的检测,第五届全国图像图形学学术会议论文集,1990.
[49] G. Burdea, J. Zhuang, Dextrous Telerobotics with force feedback——an overwiew, part 1: Human factors, Robotica, Vol. 9, 171~178, UK, 1991.
[50] 阮秋琦,计算机视觉环境及软件包的研究,第六届全国图像图形学学术会议论文集,1992
[51] Qiuqi Ruan, The Realization and Application of a Fast Searching Algorithm of the Correlation Matching, IEEE ICCT'92, 1992.
[52] Qiuqi Ruan, Yuan Baozong, Super Intelligent Visual Auditory Information Processing System, IEEE SMC', Chicago, 1992.
[53] Qiuqi Ruan, Landslide Hazard Remote Sensing Information System, 12Th Asian Conf. on Remote Sensing, 1991.
[54] 阮秋琦,图像处理和计算机视觉检测技术在美国,北方交通大学学报,16 卷, 3 期, 1992.
[55] P. Appino, B. Lewis, L. Koved, P. Ling, D. Rabenhorse, C. Codella, An Architecture for Virtual Worlds, Presence Teleoperators and Virtual Environments, Vol. 1, No. 1:1~17, 1992.
[56] M. Bajura, H. Fuchs, R. Ohbuch, Merging Virtual Objects with the Real World: See Ultrasound Imagery Within the Patient, Computer Graphics, Vol. 26, No. 2:203~210, 1992.
[57] D. Begault, The virtual reality of 3-D sound, Proceedings of Cyberarts conference, 79 ~ 87, Pasadena, CA, 1992.
[58] J. Adam, Virtual Reality is for Real, IEEE Spectrum, 22~29, 1993.
[59] J. Airey, J. Rohlf, F. Brooks Jr. Towards Image Reality with Interactive Update Rates in Complex Virtual Building Environments, Computer Graphics, AM, Vol. 24, No. 2:41~50, 1990.
[60] D. Aley, M. Donahae, A Sourceless Orientation Sensor, Sensor, 55, 1993.
[61] A. Bejczy, State-Of-the-Art in Remote Manipulation Using Virtual Environment Display, IEEE

Workshop on Force Display on Virtual Environments and Its Application to Robotic Teleoperation, Atlanta, GA 1993.

[62] 阮秋琦,模式识别与分类的发展动向,北方交通大学学报, 17 卷, 2 期,1993.

[63] 阮秋琦,智能信息处理技术在高速铁路中的应用初探,高速铁路文集,1993.

[64] 阮秋琦,相关匹配的快速算法研究,北方交通大学学报, 18 卷, 2 期,1993.

[65] 阮秋琦,计算机听视觉信息处理的现状及发展,电信科学, 9 卷, 2 期,1993.

[66] Qiuqi Ruan, The Research of the Computer Vision Information Processing Environment, ICSP'93, International Conference, 1993.

[67] 唐四春,阮秋琦, 分类分层遥感图像数据库, 第六届全国语音图像通讯信号处理学术会议论文集, 1993.

[68] 冯为民,阮秋琦,一种利用视觉特性和小波变换的混合编码方案,第六届全国语音图像通讯信号处理学术会议论文集, 1993.

[69] W. B. Thompson, P. Lechleider, E R. Stuck, Detecting Moving Objects Using the Rigidity Constraint, IEEE Trans. PAMI,Vol. 15,No. 2,ferbrury, 162~166,1993.

[70] Don Murray and Basu, Member,IEEE. Motion Tracking with an Active Camera. IEEE, Trans. on PAMI, Vol. 16, No. 5, 449~495, 1994.

[71] 阮秋琦,利用人工地震数据进行地质构造的三维重建研究,第七届全国图像图形学学术会议论文集, 1994.

[72] 阮秋琦,图像处理和计算机视觉技术在铁路现代化建设中的应用,北方交通大学学报, 20 卷, 1 期, 1996.

[73] 阮秋琦,计算机及通信中的多媒体技术,电子学报, 16 卷, 4 期, 1994.

[74] 阮秋琦,遥感图像三维重建算法研究,铁道学报, 22 卷, 11 期, 1994.

[75] 阮秋琦,智能视觉信息处理系统研究, IIT'95, 1995.

[76] 阮秋琦,地质构造三维重建研究,IIT'95, 1995.

[77] 阮秋琦, 智能视听信息处理系统研究, PC WORLD 特刊, 1995.

[78] Qiuqi Ruan, Research of the Image Coding Based on the Wavelet Transformation, IEEE, SIP'95, Las Vigas, U. S. A., 1995.

[79] 阮秋琦,信号与信息处理技术在现代通信中的作用, 全国铁路通信学术会议,1995.

[80] 阮秋琦,信息高速公路与智能信息处理技术,电子科技导报, 9 卷,9 期,1995.

[81] 阮秋琦,铁路新知识手册(信息技术部分), 中国铁道出版社,1995.

[82] T. Ishida, R. E. Korf, Moving-Target Search: A Real-Time Search for Changing Goals. IEEE Trans. PAMI, Vol. 17, No. 6: 609~619, 1995.

[83] 阮秋琦,听视觉信息处理的新策略, 北方交通大学学报, 20 卷, 2 期, 1996.

[84] Qiuqi Ruan, Wavelet Transformation Image Coding Based on the Human Visual Specificity Icsp'96. Beijing, 1996.

[85] 阮秋琦,序列图像中运动目标的检测, 铁道学报,18 卷, 6 期,1996.

[86] 孙方立,阮秋琦,基于霍夫变换的断层地质构造识别研究,铁路航测,2 期,1997.

[87] Qiuqi Ruan, Research of Tracking of Moving Target,Germany, APPT'97, International Conference, 1997.

[88] 寇元元,阮秋琦,多媒体视频卡的应用研究,海峡两岸交通大学电子信息技术研讨会论文集,1998.

[89] 阮秋琦,基于 LAN/INTERNET 的 CSCW 系统研究,海峡两岸交通大学电子信息技术研讨会论文集,1998.

[90] Quqi Ruan, New Adaptive Tracking Algorithm of Non-Rigid Object, IEEE, ICSP'98, 1998.

[91] Qiuqi Ruan, MFSC Algorithm Research of the Sequence Image Coding, IEEE JCIS'98, 1998.

[92] 阮秋琦，基于 Hausdorff 距离的非刚体目标自适应轮廓跟踪，通信学报，19 卷，6 期，1998.

[93] 阮秋琦，图像编码中的运动补偿研究，1997 通信学术交流论文集，人民邮电出版社，1997.

[94] 张颖，阮秋琦，一种基于小波变换的心电图 QRS 综合波自动检测算法，1997 通信学术交流论文集，人民邮电出版社，1997.

[95] 孙方立，阮秋琦，基于 TCP/IP 协议下的 CSCW 系统及其相关技术研究，铁道学报，20 卷，4 期，1998.

[96] 阮秋琦，序列图像编码的 FMSC 算法研究，铁道学报，21 卷，3 期，1999.

[97] 王海滨，阮秋琦，PC 机虚拟现实系统研究，信号处理，15 卷，103 期，1999.

[98] 王海滨，阮秋琦，视觉方法在虚拟现实系统头部姿态检测中的应用研究，信号处理，15 卷，106 期，1999.

[99] Wang haibin, Ruan Qiuqi, A VR Application Based PC System with Vision Tracking, APCC/OECC'99, Beijing, 1043～1047, 1999.

[100] 王海滨，阮秋琦，立体显示与视觉跟踪方法集成在虚拟现实系统中的应用研究，中国智能自动化学术会议，1999.

